Optimal Prediction Using Expert Advice and Randomized Littlestone Dimension

Yuval Filmus[1,2], Steve Hanneke[3], Idan Mehalel[1], and Shay Moran[2,1,4]

[1]The Henry and Marilyn Taub Faculty of Computer Science, Technion, Israel
[2]Faculty of Mathematics, Technion, Israel
[3]Department of Computer Science, Purdue University, USA
[4]Google Research, Israel

November 8, 2022

Abstract

A classical result in online learning characterizes the optimal mistake bound achievable by deterministic learners using the Littlestone dimension (Littlestone '88). We prove an analogous result for randomized learners: we show that the optimal *expected* mistake bound in learning a class $\mathcal{H}$ equals its *randomized Littlestone dimension*, which we define as follows: it is the largest d for which there exists a tree shattered by $\mathcal{H}$ whose *average* depth is $2d$. We further study optimal mistake bounds in the agnostic case, as a function of the number of mistakes made by the best function in $\mathcal{H}$, denoted by k. Towards this end we introduce the k-Littlestone dimension and its randomized variant, and use them to characterize the optimal deterministic and randomized mistake bounds.

As an application of our theory, we revisit the classical problem of prediction using expert advice: about 30 years ago Cesa-Bianchi, Freund, Haussler, Helmbold, Schapire and Warmuth studied prediction using expert advice, provided that the best among the n experts makes at most k mistakes, and asked what are the optimal mistake bounds (as a function of n and k). Cesa-Bianchi, Freund, Helmbold, and Warmuth ['93, '96] provided a nearly optimal bound for deterministic learners, and left the randomized case as an open problem. We resolve this question by providing an optimal learning rule in the randomized case, and showing that its expected mistake bound equals half of the deterministic bound, up to negligible additive terms. This improves upon previous works by Abernathy, Langford, and Warmuth ['06] and Brânzei and Peres ['19], which handled the regime $k \ll \log n$. In contrast, our result applies to all n and k, and reveals a threshold phenomenon in the optimal mistake bound when $k \ll \log n$ versus $k \gg \log n$.

Our proofs imply optimal learning rules, which can be seen as natural variants of the Standard Optimal Algorithm (SOA) of Littlestone: a weighted variant in the agnostic case, and a probabilistic variant in the randomized case. We conclude the paper with suggested direction for future research and open questions.

Contents

1 Introduction

A recurring phenomenon in learning theory is that different notions of learnability are captured by combinatorial parameters. Notable examples include the Vapnik–Chervonenkis (VC) dimension which characterizes PAC learnability [VC74, BEHW89] and the Littlestone dimension which characterizes online learnability [Lit88, BDPSS09]. Other examples include the Daniely–Shalev-Shwartz and Natarajan dimensions in multiclass PAC learning [Nat89, DSS14, BCD$^+$22], the star number, disagreement coefficient, and inference dimension in interactive learning [Han14, HY15, KLMZ17], the statistical query dimension in learning with statistical queries [Fel17], the representation dimension, one-way communication complexity, and Littlestone dimension in differentially private learning [FX15, BNS19, ABL$^+$22], and others.

One of the simplest and most appealing characterizations is that of online learnability by the Littlestone dimension. In his seminal work, Nick Littlestone proved that the optimal mistake-bound in online learning a class $\mathcal{H}$ is *exactly* the Littlestone dimension of $\mathcal{H}$ [Lit88]. Thus, not only does the Littlestone dimension qualitatively captures online learnability, it also provides an exact quantitative characterization of the best possible mistake bound. This distinguishes the Littlestone dimension from other dimensions in learning theory, which typically only provide asymptotic bounds on the learning complexity.

However, the exact quantitative characterization of the optimal mistake bound by the Littlestone dimension applies only in the noiseless *realizable* setting and only for *deterministic* learners. In particular, it does not apply in the more general and well-studied setting of *agnostic* online learning. The reason it does not apply is twofold: (i) because the agnostic setting allows for non-realizable sequences, and (ii) because randomized learners are in fact necessary.[1] This suggests the following question, which guides this work:

> Is there a natural dimension which captures the optimal expected mistake bound in learning a class $\mathcal{H}$ using randomized learners? How about the agnostic setting when there is no $h \in \mathcal{H}$ which is consistent with input data?

The main contribution of this work formalizes and proves affirmative answers to these questions.

Organization. In the next section we present the main results of this work. Then, in Section 3 we provide a short technical overview, where we outline the main ideas we use in our proofs. The remaining sections contain the complete proofs.

2 Main results

This section assumes familiarity with standard definitions and terminology from online learning. We refer the unfamiliar reader to Section 4, which introduces the online learning model and related basic definitions in a self-contained manner.

2.1 Realizable Case

In his seminal work from 1988, Nick Littlestone studied the optimal mistake bound in online learning a hypothesis class $\mathcal{H}$ by deterministic learning rules in the realizable setting [Lit88]; that is, under the assumption that the input data sequence is consistent with a function $h \in \mathcal{H}$.

[1] Randomized learners are necessary in the following sense: any agnostic online learner for a class $\mathcal{H}$ must be randomized, provided that $\mathcal{H}$ contains at least two functions [Cov65], see also [SSBD14, Chapter 21.2].

Littlestone dimension. Let $\mathcal{X}$ be the domain, and let $\mathcal{H}$ be a class of "$\mathcal{X} \to \{0,1\}$" predictors. The Littlestone dimension of $\mathcal{H}$, denoted $\mathtt{L}(\mathcal{H})$, is the maximal depth of a decision tree T which is shattered by $\mathcal{H}$. That is, a decision tree T whose nodes are associated with points from $\mathcal{X}$ and whose edges are associated with labels from $\{0,1\}$ such that each of the branches (root-to-leaf paths) in T is realized by some $h \in \mathcal{H}$.

Littlestone proved that the optimal mistake bound achievable by deterministic learners equals the Littlestone dimension:

Theorem 2.1 (Deterministic Mistake Bound [Lit88])**.** *The optimal <u>deterministic</u> mistake bound in online learning $\mathcal{H}$ in the realizable setting is equal to its Littlestone dimension,* $\mathtt{L}(\mathcal{H})$*.*

Littlestone further described a natural deterministic learning rule, which he dubbed the *Standard Optimal Algorithm* (SOA), that makes at most $\mathtt{L}(\mathcal{H})$ mistakes on every realizable input sequence.

Randomized Littlestone dimension. Our first main result shows that a natural probabilistic variant of the Littlestone dimension characterizes the optimal expected mistake bound for randomized learners.

Definition 2.2 (Randomized Littlestone Dimension)**.** Let T be binary tree, and consider a random walk on T that starts at the root, goes to the left or right child with probability $1/2$, and continues recursively in the same manner until reaching a leaf. Let E_T denote the expected length of a random branch which is produced by this process.
The *randomized Littlestone dimension* of a class $\mathcal{H}$, denoted by $\mathtt{RL}(\mathcal{H})$, is defined by

$$\mathtt{RL}(\mathcal{H}) = \frac{1}{2} \sup_{T \text{ shattered}} E_T.$$

To compare the randomized Littlestone dimension with the Littlestone dimension, notice that the Littlestone dimension is equal to $\sup\{m_T : T \text{ shattered}\}$, where m_T is the minimum length of a branch in T. Thus, the difference is that in $\mathtt{RL}(\mathcal{H})$ we take the expected depth rather than the minimal depth, and multiply by a factor of $1/2$.[2]

Theorem 2.3 (Main Result (i): Randomized Mistake Bound)**.** *The optimal <u>randomized</u> mistake bound in online learning $\mathcal{H}$ in the realizable setting is equal to its randomized Littlestone dimension,* $\mathtt{RL}(\mathcal{H})$*.*

We also provide an optimal randomized learning rule which can be seen as a probabilistic adaptation of Littlestone's classical SOA algorithm. See Section 3.1 for a brief overview, and Section 5.1.1 for the proof.

2.2 Agnostic Case

We next consider the agnostic setting, in which we no longer assume that the input sequence of examples is consistent with $\mathcal{H}$. Our second main result characterizes the optimal expected mistake bound in this setting.

[2] From a learning theoretic perspective it is easy to see that $\mathtt{RL}(\mathcal{H}) \leq \mathtt{L}(\mathcal{H})$, because randomized learners are more general than deterministic ones. Interestingly, this inequality is less obvious from a combinatorial perspective: indeed, for every fixed tree T we have that $E_T \geq m_T$ (because the expected length of a branch is at least the minimal length.), but it is not a priori clear why the inequality is reversed when E_T is replaced by $E_T/2$ and we take supremum over all shattered trees. See Section 5.3.2 for more details.

A common approach for handling the agnostic case is to assume a *bounded horizon* and analyze the *regret*. That is, it is assumed that the length of the input sequence (called the *horizon*) is a given parameter $\mathbf{T} \in \mathbb{N}$, and the goal is to design learning rules whose mistake bound is competitive with that of the best $h \in \mathcal{H}$ up to an additive term which is negligible in $\mathbf{T}$ (this term is called the *regret* of the algorithm).

The bounded horizon assumption simplifies the design of learning rules, by allowing them to depend on $\mathbf{T}$. A notable example is the celebrated *Multiplicative Weights* (MW) learning rule, whose learning rate depends on $\mathbf{T}$. This assumption can then be lifted by standard *doubling tricks*.[3]

The k-realizable setting. In this work we consider an alternative approach: instead of assuming a bound $\mathbf{T}$ on the horizon, we assume a bound k on the number of mistakes made by the best function in the class. Notice that this assumption can also be lifted by suitable doubling tricks as we demonstrate in Section 2.4, where we also extend our results to the bounded-horizon setting.

The upshot of this approach is that it allows for a precise combinatorial characterization of the optimal mistake bound via a natural generalization of the Littlestone dimension.

2.2.1 k-Littlestone Dimension

Let $\mathcal{H}$ be a hypothesis class, and let $k \in \mathbb{N}$. A sequence of examples $S = \{(x_i, y_i)\}_{i=1}^t$ is *k-realizable* by $\mathcal{H}$ if there exists $h \in \mathcal{H}$ such that $h(x_i) \neq y_i$ for at most k indices i. In the k-realizable setting we assume that the input sequence given to the learner is k-realizable. Notice that the case $k = 0$ amounts to realizability by $\mathcal{H}$. We say that a decision tree is *k-shattered by $\mathcal{H}$* if every branch is k-realizable by $\mathcal{H}$. The corresponding deterministic and randomized k-Littlestone dimensions of a class $\mathcal{H}$ are

$$\mathtt{L}_k(\mathcal{H}) = \sup_{T\ k\text{-shattered}} m_T \quad \text{and} \quad \mathtt{RL}_k(\mathcal{H}) = \frac{1}{2} \sup_{T\ k\text{-shattered}} E_T.$$

Theorem 2.4 (Main Result (ii): k-Realizable Mistake Bounds)**.** *Let $\mathcal{H}$ be a hypothesis class.*

1. *The optimal deterministic mistake bound in online learning $\mathcal{H}$ in the k-realizable setting equals its k-Littlestone dimension, $\mathtt{L}_k(\mathcal{H})$.*
2. *The optimal randomized mistake bound in online learning $\mathcal{H}$ in the k-realizable setting equals its k-randomized Littlestone dimension, $\mathtt{RL}_k(\mathcal{H})$.*

We also provide optimal learning rules which can be seen as weighted variants of Littlestone's classical SOA algorithm. See Section 3.1 for a brief overview and Section 7 for the proof.

As a consequence of this perspective, we are also able to establish new bounds on the optimal mistake bounds. In particular, we have the following theorem (proven in Section 7.4), which refines the regret bound proven in the recent work of [ABED$^+$21] (replacing a time horizon $\mathbf{T}$ with the number of mistakes k).

Theorem 2.5. *The optimal randomized mistake bound in online learning $\mathcal{H}$ in the k-realizable setting is upper-bounded by $k + O\left(\sqrt{k \cdot \mathtt{L}(\mathcal{H})} + \mathtt{L}(\mathcal{H})\right)$.*

[3] E.g. start by running the algorithm with $\mathbf{T} = 2$, and double $\mathbf{T}$ when reaching the $(\mathbf{T} + 1)$'st example.

2.3 Prediction Using Expert Advice

In this section, we consider the problem of *prediction using expert advice* [Vov90, LW94]. This problem studies a repeated guessing game between a learner and an adversary. In each round of the game, the learner needs to guess the label that the adversary chooses. In order to do so, the learner can use the advice of n experts. Formally, each round i in the game proceeds as follows:

(i) The experts present predictions $\hat{y}_i^{(1)}, \dots, \hat{y}_i^{(n)} \in \{0,1\}$.

(ii) The learner predicts a value $p_i \in [0,1]$.

(iii) The adversary reveals the true answer $y_i \in \{0,1\}$, and the learner suffers the loss $|y_i - p_i|$.

The value p_i should be understood as the probability (over the learner's randomness) of predicting $y_i = 1$. Notice that the adversary only gets to see p_i, which reflects the assumption that the adversary does not know the learner's internal randomness. Notice also that the suffered loss $|y_i - p_i|$ exactly captures the probability that the learner makes a mistake. The above is a standard way to model randomized learners in online learning, see e.g. [Sha12, Haz19, CBL06]. If $p_i \in \{0,1\}$ for all i, then the learner is *deterministic*, in which case $|y_i - p_i|$ is the binary indicator for whether the learner made a mistake.

We focus here on the k-realizable setting, which was suggested by [CBFHW96, CBFH$^+$97] and further studied by [ALW06, MS10, BP19]. Here, the adversary must choose the answers so that at least one of the experts makes at most k mistakes. That is, there must exist an expert j such that $y_i \neq \hat{y}_i^{(j)}$ for at most k many indices i.

The goal is to determine the optimal loss of the learner as a function of n and k. Let $\mathtt{M}^\star_D(n,k)$ denote the optimal loss of a deterministic learner and $\mathtt{M}^\star(n,k)$ denote the optimal loss of a (possibly) randomized learner.[4]

2.3.1 Deterministic Learners

For every $n \geq 1$ and $k \geq 0$, let

$$D(n,k) = \max\left\{ d : d \leq \log n + \log \binom{d}{\leq k} \right\}.$$

The value of $D(n,k)$ plays a central role in the problem of prediction using expert advice; Cesa-Bianchi et al. [CBFHW96] have shown that

$$\mathtt{M}^\star_D(n,k) = (1+o(1))D(n,k).$$

Theorem 2.6 (Bounds for Deterministic Predictors[CBFHW96]). *For every $n \geq 1$ and $k \geq 0$,*

$$D(n,k) - O(\log D(n,k)) \leq \mathtt{M}^\star_D(n,k) \leq D(n,k)$$

The lower bound is proved by constructing a k-covering code of size n that simulates the experts. When k is fixed, it can be further improved to $\mathtt{M}^\star_D(n,k) \geq D(n,k) - c(k)$, where $c(k)$ is a constant depending on k, by constructing a better covering code [CHLL97, Theorem 12.4.3].[5] We sketch good approximations to $D(n,k)$ in Table 1.

[4] Note that we assume here that k is known to the learner and that the horizon (i.e. number of rounds in the game) might be unbounded. In Section 2.4.2 below we explain how to extend our results to the complementing cases.

[5] Specifically, we can use a direct sum of k many *Hamming codes* [Ham50].

2.3.2 Randomized Learners

The main problem left open by [CBFHW96] is determining the optimal expected mistake bound for randomized learners, $\mathtt{M}^\star(n,k)$. They pointed out that[6]

$$\frac{\mathtt{M}_D^\star(n,k)}{2} \leq \mathtt{M}^\star(n,k) \leq \mathtt{M}_D^\star(n,k), \tag{1}$$

and suggested that the lower bound is nearly tight.[7]

Nearly 10 years later, Abernathy, Langford, and Warmuth [ALW06] confirmed that the lower bound is indeed nearly tight for constant k. More precisely, they showed that $\mathtt{M}^\star(n,k) \leq \mathtt{M}_D^\star(n,k)/2 + C$ for every k and every $n \geq N(k)$, where C is a universal numerical constant (independent of n,k).

More recently, Brânzei and Peres [BP19] studied the case $k = O(\log n)$. They proved that for all $k \leq \log n/5$,

$$\log_4(n) + \log_4\binom{\log n}{k} - O(k) \leq \mathtt{M}^\star(n,k) \leq \log_4(n) + k\Big(1+\frac{2k}{\ln n}\Big)\log_4\Big(\frac{\log(n)}{k}\Big) + O(k).$$

This implies that $\mathtt{M}^\star(n,k) \leq (\frac{1}{2}+o(1))D(n,k)$ for $k = o(\log n)$. The result by [BP19] applies even in the more challenging multiclass setting where the expert predictions are chosen from an arbitrary (finite) set.

In the next theorem we remove the assumption that $k \ll \log n$; we show that for all n,k

$$\mathtt{M}^\star(n,k) = \Big(\frac{1}{2} \pm o(1)\Big)D(n,k).$$

Together with Theorem 2.6, this implies that $\mathtt{M}^\star(n,k) = (\frac{1}{2}+o(1))\mathtt{M}_D^\star(n,k)$, meaning that the lower bound in Equation 1 is nearly tight, and thus resolving the question raised by [CBFHW96].

[6] The upper bound is trivial because randomized learners are more general, and the lower bound follows by rounding the randomized predictions as in Proposition 5.17

[7] One might be tempted to interpret the above inequality as implying that $\mathtt{M}^\star(n,k)$ and $\mathtt{M}_D^\star(n,k)$ are nearly the same. However, the multiplicative gap of 1/2 can be significant. For example, a randomized learner with a non-trivial error rate of 25% corresponds to a deterministic learner with 50% error-rate. The latter is trivially achieved by a random guess. For the same reason, in general sublinear regret guarantees can only be achieved by randomized learners, although they are "just" a factor of 1/2 better than deterministic learners, see e.g. [CBL06, Sha12, Haz19].

Regime	Approximation
$k = o(\log n)$	$D(n,k) \approx \log n + k \log\left(\frac{\log n}{k}\right)$
$k = \frac{\log n}{c}$ for constant c	$D(n,k) \approx k/f^{-1}(c)$
$k = \omega(\log n)$	$D(n,k) \approx 2k + 2\sqrt{k \ln n}$

Table 1: Approximations of $D(n,k)$ in various regimes

Theorem 2.7 (Main Result (iv): Bounds for Randomized Predictors)**.** *Let $\mathtt{M}^\star(n,k)$ denote the optimal expected mistake bound for prediction using expert advice in the k-realizable setting.*

1. **Small k.** *Let $n \geq 2$, and suppose that $k \leq c \log n$ for some $c < 1/2$. Then there exists a constant C, depending only on c, such that*

$$\mathtt{M}^\star(n,k) \leq \frac{D(n,k)}{2} + C.$$

2. **Large k.** *For all $n \geq 2$ and $k \geq 0$,*

$$\mathtt{M}^\star(n,k) \leq \frac{D(n,k)}{2} + O(\sqrt{D(n,k)}).$$

Moreover, the error term is tight for $n = 2$:

$$\mathtt{M}^\star(2,k) = \frac{D(2,k)}{2} + \Omega(\sqrt{D(2,k)}).$$

In particular, for all[a] *$n \geq 2$ and all k:*

$$\mathtt{M}^\star(n,k) = \left(\frac{1}{2} + o(1)\right)\mathtt{M}^\star_D(n,k),$$

where $\mathtt{M}^\star_D(n,k)$ is the optimal deterministic mistake bound for prediction using expert advice.

[a] For $n = 1$ it holds that $\mathtt{M}^\star(1,k) = \mathtt{M}^\star_D(1,k) = k$ for all k.

We note that the first item quantitatively improves the results of [ALW06] for large n: it shows that it suffices to have n exponential in k in order to get a bound of the form $D(n,k)/2 + O(1)$.

All of our bounds are attained using the randomized k-Littlestone dimension. We prove the upper bounds in Section 8.2, and the lower bound in Section 8.3. In Section 8.5 we further describe an efficient randomized algorithm in the perfect expert setting ($k = 0$), and in Section 8.6 we prove that any optimal learning rule must necessarily be improper in the sense that it cannot always predict using convex combinations of the n experts.

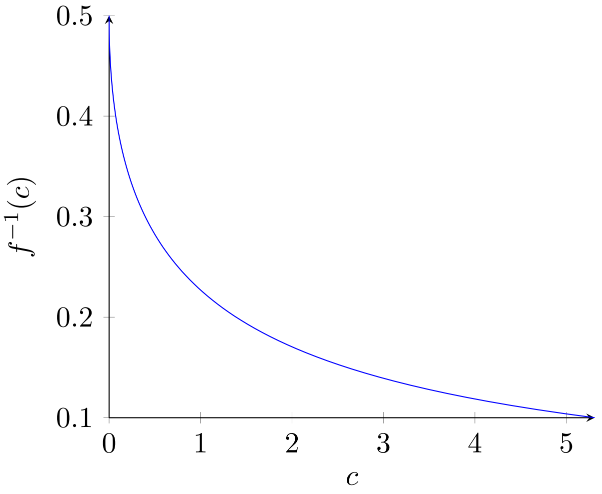

Figure 1: Plot of $f^{-1}(c)$, where $f(p) = (1 - h(p))/p$

2.4 Variations

2.4.1 Bounded Horizon

Consider learning $\mathcal{H}$ in the k-realizable setting, and let $\mathtt{M}^\star_k = \mathtt{M}^\star_k(\mathcal{H})$ denote the optimal expected mistake bound. In particular, this means that the adversary can force $\mathtt{M}^\star_k$ mistakes in expectation on any randomized learner. This would be tolerable if in order to do so the adversary must use many examples, say $1000\mathtt{M}^\star_k$. Indeed, this would mean that the learner makes only one mistake per a thousand examples (amortized), which is rather good.

This raises the question to what extent does $\mathtt{M}^\star_k$ capture the optimal mistake bound under the additional assumption that the horizon is bounded by a given $\mathbf{T} \in \mathbb{N}$. A bounded horizon is often assumed in the online learning literature, and in fact this question was explicitly asked by [CBFH+97] in the special case of prediction using expert advice.

Let $\mathtt{M}^\star_k(\mathbf{T})$ denote the optimal expected mistake bound in the k-realizable setting with horizon bounded by $\mathbf{T}$. The following result shows that $\mathtt{M}^\star_k$ provides an excellent approximation of $\mathtt{M}^\star_k(\mathbf{T})$; in particular, the scenario described above is impossible.

Theorem 2.8 (Main Result (iii): Bounded vs Unbounded Horizon)**.** *Let $\mathcal{H}$ be a hypothesis class. Let $\mathtt{M}^\star_k$ denote the optimal expected mistake bound in online learning $\mathcal{H}$ in the k-realizable setting, and let $\mathtt{M}^\star_k(\mathbf{T})$ denote the optimal expected mistake bound under the additional assumption that the input sequence has length at most $\mathbf{T}$. Then,*

1. **Long Horizon.** *If $\mathbf{T} > 2\mathtt{M}^\star_k$ then*

$$\mathtt{M}^\star_k \geq \mathtt{M}^\star_k(\mathbf{T}) \geq \mathtt{M}^\star_k - 8\exp\bigl(-(\mathbf{T} - 2\mathtt{M}^\star_k)\bigr).$$

2. **Short Horizon.** *If $\mathbf{T} \leq 2\mathtt{M}^\star_k$ then*

$$\frac{\mathbf{T}}{2} \geq \mathtt{M}^\star_k(\mathbf{T}) \geq \frac{\mathbf{T}}{2} - \sqrt{8\mathbf{T}\ln\mathbf{T}} - 1.$$

Moreover, if $\mathbf{T} \leq 2\mathtt{M}^\star_k - \sqrt{8\mathtt{M}^\star_k \ln \mathtt{M}^\star_k}$ then $\mathtt{M}^\star_k(\mathbf{T}) \in [\frac{\mathbf{T}}{2} - 1, \frac{\mathbf{T}}{2}]$, and if $\mathbf{T} \leq \mathtt{M}^\star_k$ then $\mathtt{M}^\star_k(\mathbf{T}) = \frac{\mathbf{T}}{2}$.

That is, once $\mathbf{T} > 2\mathtt{M}^\star_k$ we see that $\mathtt{M}^\star_k(\mathbf{T})$ converges to $\mathtt{M}^\star_k$ exponentially fast: e.g. $\mathtt{M}^\star_k(\mathbf{T}) \in [\mathtt{M}^\star_k - 0.001, \mathtt{M}^\star_k]$ whenever $\mathbf{T} \geq 2\mathtt{M}^\star_k + 10$. In the complementary case, when $\mathbf{T} \leq 2\mathtt{M}^\star_k$ we see that $\mathtt{M}^\star_k(\mathbf{T}) \approx \frac{\mathbf{T}}{2}$; notice that $\frac{\mathbf{T}}{2}$ is the largest possible value that the optimal expected mistake bound can attain: indeed, a randomized learner which always predicts a random label achieves an expected mistake bound of $\frac{\mathbf{T}}{2}$.

Our proof of Theorem 2.8 appears in Section 6.3 (long horizon) and in Section 6.4 (short horizon). The proof relies on a simple extension of our characterization to this setting: consider the following modification of the Littlestone dimension and its randomized variant:

$$\mathtt{L}_k(\mathcal{H}, \mathbf{T}) = \sup_{\substack{T \text{ shattered} \\ \mathsf{depth}(T) \leq \mathbf{T}}} m_T \quad \text{and} \quad \mathtt{RL}_k(\mathcal{H}, \mathbf{T}) = \frac{1}{2} \sup_{\substack{T \text{ shattered} \\ \mathsf{depth}(T) \leq \mathbf{T}}} E_T.$$

The bounded randomized Littlestone dimension gives the precise mistake bound in this setting:

Theorem 2.9 (Optimal Mistake Bounds: Bounded Horizon)**.** *Let $\mathcal{H}$ be an hypothesis class.*

1. *The optimal deterministic mistake bound in online learning $\mathcal{H}$ in the k-realizable setting with horizon $\mathbf{T}$ equals its bounded k-Littlestone dimension, $\mathtt{L}_k(\mathcal{H}, \mathbf{T})$.*[8]

2. *The optimal randomized mistake bound in online learning $\mathcal{H}$ in the k-realizable setting with horizon $\mathbf{T}$ equals its bounded k-randomized Littlestone dimension, $\mathtt{RL}_k(\mathcal{H}, \mathbf{T})$.*

We prove Theorem 2.9 in Section 6.1.

Prediction using Expert Advice. Also the problem of prediction using expert advice is often considered when the number of rounds is bounded (e.g. [CBFH$^+$97]). Let $\mathtt{M}^\star(n, k, \mathbf{T})$ be the optimal loss of the learner when the number of rounds is $\mathbf{T}$. By a simple reduction to Theorem 2.9 we show that

$$\mathtt{M}^\star(n, k, \mathbf{T}) \approx \begin{cases} \mathtt{M}^\star(n, k) & \text{if } \mathbf{T} \geq 2\mathtt{M}^\star(n, k), \\ \frac{\mathbf{T}}{2} & \text{if } \mathbf{T} < 2\mathtt{M}^\star(n, k). \end{cases}$$

The exact bounds are as in Theorem 2.9 when replacing $\mathtt{M}^\star(n, k, \mathbf{T})$ and $\mathtt{M}^\star(n, k)$ with $\mathtt{M}^\star_k(\mathbf{T})$ and $\mathtt{M}^\star_k$.

2.4.2 Adaptive Algorithms

The analysis in much of this work considers the case where the learning algorithm may depend explicitly on a bound k on the number of mistakes of the best hypothesis (or expert). However, it is also desirable to study mistake bounds achievable *adaptively*: that is, by a single algorithm that applies to all k. We present here one simple approach to obtaining such an algorithm, with a corresponding mistake bound. However, the bound we obtain may likely be improvable, and generally we leave the question of obtaining a tightest possible adaptively-achievable mistake bound as an open problem.

Theorem 2.10. *There is an adaptive algorithm (i.e., which has no knowledge of k^*) such that, for every k^*-realizable sequence for $\mathcal{H}$, its expected number of mistakes is at most*

$$\mathtt{M}^\star_{k^*} + O\left(\sqrt{\mathtt{M}^\star_{k^*} \log\big((k^* + 1) \log \mathtt{M}^\star_{k^*}\big)}\right).$$

[8] Trivially, $\mathtt{L}_k(\mathcal{H}, \mathbf{T}) = \min\{\mathbf{T}, \mathtt{L}_k(\mathcal{H})\}$.

In the special case of the general *experts* setting, since we know that $\mathtt{M}^\star(n,k^*) = \Omega(k^*+\log(n))$, we obtain the following bound on the expected number of mistakes:

$$\mathtt{M}^\star(n,k^*) + O\Big(\sqrt{\mathtt{M}^\star(n,k^*)\log \mathtt{M}^\star(n,k^*)}\Big) = (1+o(1))\mathtt{M}^\star(n,k^*).$$

In particular, combining this with Theorems 2.6 and 2.7, we find that this algorithm adaptively still achieves an expected number of mistakes $\left(\frac{1}{2}+o(1)\right)\mathtt{M}^\star_D(n,k^*)$.

On the other hand, in the case of concept classes $\mathcal{H}$ with a bounded Littlestone dimension $\mathtt{L}(\mathcal{H})$, we know from Theorem 2.5 that

$$\mathtt{M}^\star_{k^*} \le k^* + O\Big(\sqrt{k^*\mathtt{L}(\mathcal{H})} + \mathtt{L}(\mathcal{H})\Big).$$

Theorem 2.10 implies that the adaptive procedure nearly preserves the form of this upper bound, guaranteeing a slightly larger bound of the form

$$k^* + O\Big(\sqrt{k^*\mathtt{L}(\mathcal{H})\log(k^*\log \mathtt{L}(\mathcal{H}))} + \mathtt{L}(\mathcal{H})\Big).$$

Our proof of Theorem 2.10 appears in Section 7.5. The adaptive technique we propose involves using an experts algorithm of [KvE15] named Squint, with experts defined by the optimal randomized algorithm for the k-realizable setting, for all values of k.

3 Technical Overview

In its greatest generality, online prediction is a game involving two randomized parties, an adversary who is producing examples, and a learner who is trying to correctly predict the labels of all or most of these examples. In the realizable case, the adversary is moreover constrained by a hypothesis class which must be adhered to.

Various techniques are used in the literature to analyze this sophisticated setting. On the one hand, learning rules show which hypothesis classes lend themselves to learning, and on the other hand, strategies for the adversary put limitations on what can be learned, and at what cost.

In this work, we identify the combinatorial core behind many settings of online learning. In this, we follow up on Nick Littlestone's classical work on deterministic online learning, as well as on other classical work in learning theory such as that the foundational work of Vapnik and Chervonenkis.

Reducing the messy probabilistic setting of online learning to the clean combinatorial setting of shattered trees enables us to tackle open questions about prediction using expert advice, which are hard to approach directly.

3.1 Combinatorial Characterizations

The Littlestone dimension of a hypothesis class $\mathcal{H}$ is the maximal depth of a complete binary tree which is shattered by $\mathcal{H}$. A tree of depth D easily translates into a strategy for the adversary which forces the learner to make D mistakes. In other words, a tree shattered by $\mathcal{H}$ is an obvious obstacle to learning $\mathcal{H}$.

The magic of Littlestone dimension is the opposite direction: Littlestone's SOA learning rule makes at most $\mathtt{L}(\mathcal{H})$ mistakes, showing that trees shattered by $\mathcal{H}$ are the *only* obstacle for learning $\mathcal{H}$. This is a common phenomenon in mathematics: an obvious necessary condition turns out to be (less obviously) sufficient.

Defining the randomized Littlestone dimension. In order to motivate the definition of the randomized Littlestone dimension, let us first examine the (deterministic) Littlestone dimension. Given a tree T shattered by $\mathcal{H}$, the adversary executes the following strategy, starting at the root:

At an internal node labelled x, ask the learner for the label of x, and follow the opposite edge.

This strategy follows a branch of T, and forces the learner to make a mistake in each round. The total number of mistakes which the adversary can guarantee is precisely m_T, the minimum length of a branch in T. The resulting input sequence is realizable by $\mathcal{H}$ since T is shattered by $\mathcal{H}$.

The definition of the randomized Littlestone dimension follows a similar approach, but uses a different strategy for the adversary:

At an internal node labelled x, ask the learner for the label of x, and follow a random *edge.*

This strategy also follows a branch of T, and it forces the learner to make *half* a mistake in each round, in expectation.[9] The total expected number of mistakes is $E_T/2$, where E_T is the expected length of a random branch of T.

We define the randomized Littlestone dimension by considering all such adversary strategies:

$$\mathtt{RL}(\mathcal{H}) = \frac{1}{2} \sup_{T \text{ shattered}} E_T.$$

The supremum is not always achieved, even if we allow infinite trees, as we demonstrate in Section 5.5.

Extending the Standard Optimal Algorithm. Littlestone's Standard Optimal Algorithm (SOA) makes at most $\mathtt{L}(\mathcal{H})$ mistakes on any realizable input sequence. The algorithm is very simple. It maintains a subset V of $\mathcal{H}$ which consists of all hypotheses which are consistent with the data seen so far. Given a sample x, one of the following must hold, where $V_{x\to y}$ is the subset of V consisting of all hypotheses assigning to x the label y:

1. $\mathtt{L}(V_{x\to 0}) < \mathtt{L}(V)$. The learner predicts $\hat{y} = 1$.

2. $\mathtt{L}(V_{x\to 1}) < \mathtt{L}(V)$. The learner predicts $\hat{y} = 0$.

One of these cases must hold, since otherwise we could construct a tree of depth $\mathtt{L}(V) + 1$ shattered by V. Each time that the learner makes a mistake, $\mathtt{L}(V)$ decreases by 1, and so the learner makes at most $\mathtt{L}(\mathcal{H})$ mistakes.

Our randomized extension of SOA, which we call $\mathsf{RandSOA}$, follows a very similar strategy. It maintains V in the same way. Given a sample x, we want to make a prediction p which "covers all bases", that is, results in a good outcome for the learner whatever the correct label y is. Given a prediction p, the adversary can guarantee a loss of

$$\max\{p + \mathtt{RL}(V_{x\to 0}), 1 - p + \mathtt{RL}(V_{x\to 1})\}.$$

For the optimal choice of p, this quantity is at most $\mathtt{RL}(V)$, as we show in Section 5.1.1.

[9] Recall that we model a randomized learner as a learner which makes a "soft" prediction $p \in [0,1]$; if the true label is y, then the learner's loss is $|p - y|$. When we choose the label y at random, the expected loss is $\mathbb{E}[|p - y|] = \frac{1}{2}$ regardless of p.

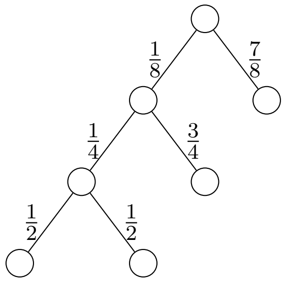

Figure 2: A quasi-balanced tree. The edges are labelled with the unique weights. The sum of weights in each branch is $\frac{7}{8}$, which is half the expected branch length $\frac{7}{4}$.

The k-realizable setting and weighted SOA. The k-realizable setting is handled similarly. In the definition of randomized Littlestone dimension, instead of requiring the tree to be shattered, it suffices for it to be k-shattered, since the adversary need only produce an input sequence which is k-realizable.

The main novelty in this setting is a *weighted* analog of the SOA learning rule. This weighted SOA rule relates to the classical SOA in a similar way like the *Weighted Majority* algorithm relates to *Halving*. In particular, it keeps track, for each hypothesis, how many more mistakes are allowed. Accordingly, we consider the more generalized setting of *weighted hypothesis classes*. These are hypothesis classes in which each hypothesis has a "mistake budget". The definition of randomized Littlestone dimension extends to this setting, and allows us to generalize RandSOA to the randomized agnostic setting.

3.2 Quasi-balanced Trees

Given a hypothesis class $\mathcal{H}$, how does an optimal strategy for the adversary look like? Such a strategy must make the analysis of RandSOA tight, and in particular, if the first sample it asks is x, then

$$\mathtt{RL}(\mathcal{H}) = p + \mathtt{RL}(\mathcal{H}_{x\to 0}) = 1 - p + \mathtt{RL}(\mathcal{H}_{x\to 1}),$$

where p is the prediction of the learner.[10]

The strategy of the adversary naturally corresponds to a tree which is shattered by $\mathcal{H}$: the root is labelled x, and the edge labelled y leads to a tree corresponding to an optimal strategy for $\mathcal{H}_{x\to y}$. Suppose that we further assign weights to the edges touching the root: the 0-edge gets the weight p, and the 1-edge gets the weight $1-p$. If we assign weights to the remaining edges recursively then the resulting tree satisfies the following property:

Every branch has the same total weight $\mathtt{RL}(\mathcal{H})$*.*

More generally, a tree T is *quasi-balanced* if we can assign non-negative weight to its edges such that (i) the weights of the two edges emanating from a vertex sum to 1, and (ii) all branches have the same total weight (which must be $E_T/2$). If a tree is quasi-balanced then the weight assignment turns out to be *unique*.

A tree in which all branches have the same depth is quasi-balanced, but the class of quasi-balanced trees is a lot richer, including for example the path appearing in Figure 2.

There is a simple criterion for quasi-balancedness:

[10] Strictly optimal strategies do not always exist, and even when they do, they might require an unbounded number of rounds. For the sake of exposition we gloss over these difficulties.

A tree T is quasi-balanced if and only if it is *monotone*: if w is a descendant of v then $E_{T_w} \leq E_{T_v}$, where T_u is the subtree rooted at u.

Since the loss guaranteed by an adversary following the strategy corresponding to a tree T is $E_T/2$, it is clear that the best strategy is always monotone. This argument shows that

$$\texttt{RL}(\mathcal{H}) = \frac{1}{2} \sup_{T \text{ shattered, monotone}} E_T.$$

In other words, it suffices to consider only quasi-balanced trees when defining the randomized Littlestone dimension. This is the randomized counterpart of a trivial property of the Littlestone dimension: in order to define the Littlestone dimension, it suffices to consider *balanced* trees, that is, trees in which all branches have the same length. We can view quasi-balancedness as a relaxation of strict balancedness.

Concentration of expected branch length. The randomized Littlestone dimension is defined in terms of the expected branch length. However, several of our results require knowledge of the distribution of the branch length.

For example, Theorem 2.8 states that $2\texttt{RL}(\mathcal{H}) + O(\log(1/\epsilon))$ rounds are needed in order for the adversary to guarantee a loss of $\texttt{RL}(\mathcal{H}) - \epsilon$. The number of rounds corresponds to the depth of the tree, and so the natural way to prove such a result would be to start with a tree T satisfying $E_T/2 = \texttt{RL}(\mathcal{H})$, and prune it to depth $2\texttt{RL}(\mathcal{H}) + O(\log(1/\epsilon))$. We would like to say that this does not reduce the expected branch length by much, since the length of most branches does not exceed E_T by much. Other applications such as prediction using expert advice need concentration from the other side (the length of most branches does not fall behind E_T by much).

It is possible to construct trees for which the length of a random branch isn't concentrated around its expectation. For example, we can take an infinite path which, every so often, splits into a deep complete binary tree. If we are careful, we can guarantee that the expected branch length is finite but its variance is infinite.

At this point, quasi-balancedness comes to the rescue. The monotonicity property of quasi-balanced trees implies that the choice of an edge at every step of a random branch does not affect the final length by much. Consequently, Azuma's inequality (a version of Chernoff's inequality for martingales) shows that for quasi-balanced trees, the length of a random branch is strongly concentrated around its expectation. This simple observation drives several of our strongest results.

3.3 Prediction using Expert Advice

At first, the setting of prediction using expert advice looks similar, but not identical, to our setting. However, it turns out that it is actually a *special case* of our setting, for a specific hypothesis class known as the *universal hypothesis class* $\mathcal{U}_n$.

The class $\mathcal{U}_n$ contains n different hypotheses, which correspond to the experts. For each possible set of predictions $\hat{y}^{(1)}, \ldots, \hat{y}^{(n)}$ there is a corresponding element in the domain. In other words, the domain is $\mathcal{X} = \{0,1\}^n$, and the hypotheses in $\mathcal{U}_n$ are the n projections $h_i(x_1, \ldots, x_n) = x_i$.

With this equivalence in place, we can apply the theory we have developed so far to analyze prediction using expert advice. Our main result concerning this setting, Theorem 2.7, consists of three different statements: two upper bounds on $\texttt{M}^\star(n,k)$, and one matching lower bound on $\texttt{M}^\star(2,k)$.

The two upper bounds are proved using a similar approach. In view of the equivalence above, we want to bound the expected branch length of any tree T which is k-shattered by $\mathcal{U}_n$. We can assume that T is quasi-balanced, and so the length of a random branch of T is roughly E_T. If T were strictly balanced, then a random branch would be k-realizable by $\mathcal{U}_n$ with probability at most

$$n\frac{\binom{E_T}{\le k}}{2^{E_T}},$$

and so $E_T \le D(n,k)$ by definition of $D(n,k)$. Since T is only quasi-balanced, we get a slightly worse bound. Quantitatively, our loss stems from the tail bound in Azuma's inequality, and delicate calculations are required to obtain the statements in Theorem 2.7.[11]

We prove the lower bound on $\mathtt{M}^\star(2,k)$ by identifying the optimal tree. Intuitively, it seems obvious that rounds in which both experts make the same prediction are "wasteful", and we can show this formally. By symmetry, we can assume that the first expert always predicts 0 and that the second expert always predicts 1. We can construct the corresponding tree explicitly, and conclude that

$$\mathtt{M}^\star(2,k) = k + \frac{(k+1/2)\binom{2k}{k}}{4^k}.$$

4 Background and Basic Definitions

Unless stated otherwise, our logarithms are base 2.

Online Learning. Let $\mathcal{X}$ be a set called the *domain*, and $\mathcal{Y}$ be a set called the *label set*. In this work we focus on *binary classification*, and thus $\mathcal{Y} = \{0,1\}$. A pair $(x,y) \in \mathcal{X} \times \mathcal{Y}$ is called an *example*, and an element $x \in \mathcal{X}$ is called an *instance* or an *unlabeled example*. A function $h\colon \mathcal{X} \to \mathcal{Y}$ is called a *hypothesis* or a *concept*. A *hypothesis class*, or a *concept class*, is a set $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$. A sequence of examples $S = \{(x_i,y_i)\}_{i=1}^t$ is said to be *realizable* by $\mathcal{H}$ if there exists $h \in \mathcal{H}$ such that $h(x_i) = y_i$ for all $1 \le i \le t$.

Online learning [SSBD14, CBFH+97] is a repeated game between a learner and an adversary. Each round i in the game proceeds as follows:

(i) The adversary sends the learner an unlabeled example $x_i \in \mathcal{X}$.

(ii) The learner predicts a value $p_i \in [0,1]$ and reveals it to the adversary.

(iii) The adversary reveals the true label y_i, and the learner suffers the *loss* $|y_i - p_i|$.

The value p_i should be understood as the probability (over the learner's randomness) of predicting $y_i = 1$. Notice that the adversary only gets to see p_i, which reflects the assumption that the adversary does not know the learner's internal randomness. Notice also that the suffered loss $|y_i - p_i|$ exactly captures the probability that the learner makes a mistake. The above is a standard way to model randomized learners in online learning, see e.g. [Sha12]. If $p_i \in \{0,1\}$ for all i, then the learner is *deterministic*, in which case $|y_i - p_i|$ is the binary indicator for whether the learner made a mistake.

We model learners as functions $\mathsf{Lrn}\colon (\mathcal{X} \times \mathcal{Y})^* \times \mathcal{X} \to [0,1]$. Given a learning rule Lrn and an input sequence of examples $S = (x_1,y_1),\dots,(x_t,y_t)$, we denote the (expected) number of mistakes Lrn makes on S by

$$\mathtt{M}(\mathsf{Lrn};S) = \sum_{i=1}^t |y_i - p_i|,$$

[11] The weaker bound $\mathtt{M}^\star(n,k) \le (\frac{1}{2} + o(1))D(n,k)$ is much easier to prove: it doesn't require Azuma's inequality (Markov's inequality suffices), and the calculations are much simpler.

where $p_i = \mathsf{Lrn}((x_1, y_1), \ldots, (x_{i-1}, y_{i-1}), x_i)$ is the prediction of the learner on the i'th example.

A hypothesis class $\mathcal{H}$ is *online learnable* (or *learnable*) if there exists a finite bound M and a learning rule Lrn such that for any input sequence S which is realizable by $\mathcal{H}$ it holds that $\mathtt{M}(\mathsf{Lrn}; S) \leq M$. We define the *optimal* randomized mistake bound of $\mathcal{H}$ to be

$$\mathtt{M}^\star(\mathcal{H}) = \inf_{\mathsf{Lrn}} \sup_{S} \mathtt{M}(\mathsf{Lrn}; S) \tag{2}$$

where the infimum is taken over all learning rules, and the supremum is taken over all realizable input sequences S.

We denote by $\mathtt{M}^\star_D(\mathcal{H})$ the optimal deterministic mistake bound of $\mathcal{H}$. That is, $\mathtt{M}^\star_D(\mathcal{H})$ is defined in the same way as $\mathtt{M}^\star(\mathcal{H})$, with the additional restriction that Lrn must be deterministic (that is, the output must be in $\{0, 1\}$).

When $\mathcal{H} = \emptyset$, the set of realizable input sequences is empty, and therefore the supremum is not defined. It is technically convenient to deal with this special case by defining $\mathtt{M}^\star_D(\emptyset) = \mathtt{M}^\star(\emptyset) = -1$. When the context is clear, we may sometimes refer to the deterministic or randomized mistake bound as the *accumulating loss* of the learner through the entire game, or simply as the learner's *loss* through the entire game.

Decision Trees and the Littlestone Dimension. In this paper, a *tree* T refers to a finite full rooted ordered binary tree (that is, a rooted binary tree where each node which is not a leaf has a left child and a right child), equipped with the following information:

1. Each internal node v is associated with an instance $x \in \mathcal{X}$.
2. For every internal node v, the left outgoing edge is associated with the label 0, and the right outgoing edge is associated with the label 1.

We stress that by default, the trees we consider are finite and their vertices are labelled. Whenever we consider infinite trees or unlabelled trees, we specifically mention these attributes.

The tree is directed from the root towards the leaves.

A *prefix* of the tree T is any path that starts at the root. In this paper, a path is defined by a sequence of consecutive vertices. If a path is not empty, we may refer it by the sequence of consecutive edges corresponding with the sequence of consecutive vertices defining it. A prefix $v_0, v_1, \ldots, v_t$ defines a sequence of examples $(x_1, y_1), \ldots, (x_t, y_t)$ in a natural way: for every $i \in [t]$, x_i is the instance corresponding to the node v_{i-1}, and y_i is the label corresponding to the edge $v_{i-1} \to v_i$. A prefix is called *maximal* if it is maximal with respect to containment, that is, there is no prefix in the tree that strictly contains it. This is equivalent to requiring that v_t be a leaf. A maximal prefix is called a *branch*, and the set of branches of T is denoted by $B(T)$. The length of a prefix is the number of edges in it (so, the length is equal to the size of the corresponding sequence of examples).

A prefix in the tree is said to be *realizable* by $\mathcal{H}$ if the corresponding sequence of examples is realizable by $\mathcal{H}$. A tree T is *shattered* by $\mathcal{H}$ if all branches in T are realizable by $\mathcal{H}$. The *Littlestone dimension* of a hypothesis class $\mathcal{H}$, denoted by $\mathtt{L} = \mathtt{L}(\mathcal{H})$, is the maximal depth of a *complete* (also known as *perfect*, or *balanced*) binary tree (that is, a tree in which all branches have the same depth) shattered by $\mathcal{H}$ if $\mathcal{H} \neq \emptyset$, and -1 when $\mathcal{H} = \emptyset$. If the maximum does not exist, then $\mathtt{L} = \infty$.

Littlestone Dimension $\equiv$ Optimal Deterministic Mistake Bound. In his seminal work from 1988, Nick Littlestone proved that the optimal mistake bound of a deterministic learner is characterized by the Littlestone dimension:

Theorem 4.1 (Optimal Deterministic Mistake Bound [Lit88])**.** *Let $\mathcal{H}$ be a hypothesis class. Then, $\mathcal{H}$ is online learnable if and only if $\mathtt{L}(\mathcal{H}) < \infty$. Further, the optimal deterministic mistake bound satisfies $\mathtt{M}^\star(\mathcal{H}) = \mathtt{L}(\mathcal{H})$.*

Doob's Exposure Martingales. Let $f\colon \{0,1\}^{\mathbb{N}} \to \mathbb{R}$. Consider the random variable $X = f(\vec{b})$, where $\vec{b}$ is sampled uniformly at random. Define a sequence $L_0, L_1, L_2, \ldots$, each defined by $L_i = \mathbb{E}[X|b_1, \ldots, b_{i-1}]$ (so $L_0 = \mathbb{E}[X]$). The sequence $L_0, L_1, L_2, \ldots$ is called an *exposure martingale.* It is well-known that an exposure martingale is indeed a martingale [Doo53].

5 Randomized Littlestone Dimension and Optimal Expected Mistake Bound

In this section we study the randomized Littlestone dimension. We start with Section 5.1, in which we define the randomized Littlestone dimension and prove that it characterizes the optimal randomized mistake bounds exactly.

The randomized Littlestone dimension is defined using trees, which correspond to strategies of the adversary. We study a special class of trees, *quasi-balanced trees*, in Section 5.2, showing that they give optimal strategies for the adversary. Several applications of quasi-balanced trees are presented in Section 5.3; more applications are found throughout the paper.

We close this section by showing how to accommodate infinite trees (Section 5.4), and by briefly discussing the issue of trees attaining the randomized Littlestone dimension exactly (Section 5.5); more discussion on the latter issue appears in Section 7.

5.1 Main Result and Proof

The first main contribution of this paper is a characterization of the optimal randomized mistake bound in terms of a combinatorial parameter we call the *randomized Littlestone dimension* and denote by $\mathtt{RL} = \mathtt{RL}(\mathcal{H})$.

We define $\mathtt{RL}(\mathcal{H})$ using a natural distribution on the branches of trees (a branch is a root-to-leaf path). Given a tree T, a *random branch* is chosen by starting at the root, and at each step, picking an edge leaving the current vertex uniformly at random, until reaching a leaf. We denote the expected length of a random branch by E_T. It is given explicitly by the formula

$$E_T = \sum_{b \in B(T)} |b| \cdot 2^{-|b|},$$

where $B(T)$ is the set of branches of T. If we think of a random branch as a distribution over $B(T)$, then E_T is its entropy.

It is convenient to define the length of the empty branch to be -1. With this convention, the expected branch length in T satisfies the recursion

$$E_T = 1 + \frac{E_{T_0} + E_{T_1}}{2}, \tag{3}$$

where T_0, T_1 are the subtrees of the root of T, which are empty when T is a leaf.

Definition 5.1 (Randomized Littlestone Dimension)**.** Let $\mathcal{H}$ be a hypothesis class. The *randomized Littlestone dimension* of $\mathcal{H}$, denoted by $\mathtt{RL}(\mathcal{H})$, is defined by

$$\mathtt{RL}(\mathcal{H}) = \frac{1}{2} \sup_{T \text{ shattered}} E_T.$$

In the special case when $\mathcal{H} = \emptyset$, define $\mathtt{RL}(\mathcal{H}) = -1$.

To compare $\mathtt{RL}(\mathcal{H})$ with $\mathtt{L}(\mathcal{H})$, let us consider the following equivalent way of defining $\mathtt{L}(\mathcal{H})$:

$$\mathtt{L}(\mathcal{H}) = \sup_{T \text{ shattered}} m_T,$$

where m_T is the minimum length of a branch in T. Thus, the difference is that in $\mathtt{RL}(\mathcal{H})$ we take the expected depth rather than the minimal depth, and multiply by a factor of $1/2$.

Theorem 5.2 (Optimal Randomized Mistake Bound)**.** *Let $\mathcal{H}$ be a hypothesis class. Then,*

$$\mathtt{M}^\star(\mathcal{H}) = \mathtt{RL}(\mathcal{H}).$$

We prove the theorem in Subsection 5.1.1 using *randomized* SOA, a randomized adaptation of Littlestone's classical SOA algorithm. This shows that the infimum in Equation (2) is realized by a minimizer.

5.1.1 Proof of Theorem 5.2

The case $\mathcal{H} = \emptyset$ holds by definition. Therefore we assume that $\mathcal{H} \neq \emptyset$. The lower bound "$\mathtt{RL}(\mathcal{H}) \leq \mathtt{M}^\star(\mathcal{H})$" boils down to the following lemma:

Lemma 5.3. *Let $\mathcal{H}$ be a hypothesis class, and let T be a finite tree which is shattered by $\mathcal{H}$. Then, for every learning rule Lrn there exists a realizable sequence S so that $\mathtt{M}(\mathsf{Lrn}; S) \geq E_T/2$. Moreover, there exists such a sequence S which corresponds to one of the branches of T.*

Proof. The proof is given by a simple probabilistic argument. Suppose that we pick a random branch in the tree according to the random branch distribution: begin at the root, pick a random child of the root uniformly at random, and recursively pick a random branch in the corresponding subtree. Consider the random variable

$$L_T = \mathtt{M}(\mathsf{Lrn}; S),$$

where S is the sequence of examples corresponding to a random branch drawn as above. It suffices to show that $\mathbb{E}[L_T] = E_T/2$. We prove this by induction on the depth of T.

In the base case, T is a single leaf, and there are no internal nodes. Hence S is always the empty sequence, and $\mathbb{E}[L_T] = 0 = E_T/2$, as required.

For the induction step, let T_0 and T_1 be the left and right subtrees of T, respectively. The expected loss of Lrn on the first example in S is $1/2$, because the label $y \in \{0, 1\}$ is chosen uniformly at random, independently of the learner's prediction (formally, $\frac{|0-p|+|1-p|}{2} = 1/2$ for all $p \in [0, 1]$). Therefore, by linearity of expectation,

$$\begin{aligned}
\mathbb{E}[L_T] &= \frac{1 + \mathbb{E}[L_{T_0}] + \mathbb{E}[L_{T_0}]}{2} \\
&= \frac{1 + E_{T_0}/2 + E_{T_1}/2}{2} && \text{(by the induction hypothesis)} \\
&= E_T/2, && \text{(by Eq. (3))}
\end{aligned}$$

as required. □

By applying Lemma 5.3 on every shattered tree and taking the supremum, we conclude the lower bound:

Corollary 5.4 (Lower bound)**.** *For every hypothesis class $\mathcal{H}$ it holds that $\mathtt{M}^\star(\mathcal{H}) \geq \mathtt{RL}(\mathcal{H})$.*

We now turn to prove the upper bound "$\mathtt{RL}(\mathcal{H}) \geq \mathtt{M}^\star(\mathcal{H})$". This is achieved via the RandSOA learning rule, described in Figure 3.

We begin with the following useful property of RL:

Observation 5.5. *Let $\mathcal{H}$ be a non-empty hypothesis class. Then,*

$$\mathtt{RL}(\mathcal{H}) = \frac{1}{2} \sup_{x \in \mathcal{X}} \big(1 + \mathtt{RL}(\mathcal{H}_{x\to 0}) + \mathtt{RL}(\mathcal{H}_{x\to 1})\big).$$

Proof. Observation 5.5 follows from Equation (3): let $\mathcal{S}(\mathcal{H})$ denote the set of trees that are shattered by $\mathcal{H}$, and for $x \in \mathcal{X}$, let $\mathcal{S}_x(\mathcal{H}) \subseteq \mathcal{S}(\mathcal{H})$ denote the set of trees that are shattered by $\mathcal{H}$ whose root is labelled by x. Then,

$$\mathtt{RL}(\mathcal{H}) = \frac{1}{2} \sup_{T \in \mathcal{S}(\mathcal{H})} E_T = \frac{1}{2} \sup_{x} \sup_{T \in \mathcal{S}_x(\mathcal{H})} E_T.$$

By Equation (3),

$$\sup_{T \in \mathcal{S}_x(\mathcal{H})} E_T = 1 + \frac{\sup_{T_1 \in \mathcal{S}(\mathcal{H}_{x\to 1})} E_{T_1} + \sup_{T_0 \in \mathcal{S}(\mathcal{H}_{x\to 0})} E_{T_0}}{2} = 1 + \mathtt{RL}(\mathcal{H}_{x\to 1}) + \mathtt{RL}(\mathcal{H}_{x\to 0}),$$

which finishes the proof. □

Notice that the classical Littlestone dimension satisfies a similar recursion:

$$\mathtt{L}(\mathcal{H}) = \sup_{x \in \mathcal{X}} \big(1 + \min\{\mathtt{L}(\mathcal{H}_{x\to 1}), \mathtt{L}(\mathcal{H}_{x\to 0})\}\big).$$

The following lemma is the crux of the analysis: it guides the choice of the prediction p_i in each round.

Lemma 5.6 (Optimal prediction for each round)**.** *Let $\mathcal{H}$ be a hypothesis class, and let $x \in \mathcal{X}$. Then there exists $p \in [0,1]$ so that*

$$p + \mathtt{RL}(\mathcal{H}_{x\to 0}) \leq \mathtt{RL}(\mathcal{H}) \quad \textit{and} \quad (1-p) + \mathtt{RL}(\mathcal{H}_{x\to 1}) \leq \mathtt{RL}(\mathcal{H}).$$

Proof of Lemma 5.6. If $\mathtt{RL}(\mathcal{H}) = \infty$ then the lemma is trivial. Therefore we assume that $\mathtt{RL}(\mathcal{H}) < \infty$. Assume first that $|\mathtt{RL}(\mathcal{H}_{x\to 0}) - \mathtt{RL}(\mathcal{H}_{x\to 1})| > 1$. If $\mathtt{RL}(\mathcal{H}_{x\to 0}) + 1 < \mathtt{RL}(\mathcal{H}_{x\to 1})$, then by choosing $p = 1$ and applying the fact that $\mathtt{RL}(\mathcal{H}') \leq \mathtt{RL}(\mathcal{H})$ if $\mathcal{H}' \subseteq \mathcal{H}$ we get

$$\begin{aligned} p + \mathtt{RL}(\mathcal{H}_{x\to 0}) &= 1 + \mathtt{RL}(\mathcal{H}_{x\to 0}) < \mathtt{RL}(\mathcal{H}_{x\to 1}) \leq \mathtt{RL}(\mathcal{H}), \\ 1 - p + \mathtt{RL}(\mathcal{H}_{x\to 1}) &= \mathtt{RL}(\mathcal{H}_{x\to 1}) \leq \mathtt{RL}(\mathcal{H}), \end{aligned}$$

as desired. The case $\mathtt{RL}(\mathcal{H}_{x\to 1}) + 1 < \mathtt{RL}(\mathcal{H}_{x\to 0})$ is treated similarly.

It remains to handle the case when $|\mathtt{RL}(\mathcal{H}_{x\to 0}) - \mathtt{RL}(\mathcal{H}_{x\to 1})| \leq 1$. Set

$$p := \frac{1 + \mathtt{RL}(\mathcal{H}_{x\to 1}) - \mathtt{RL}(\mathcal{H}_{x\to 0})}{2}.$$

By assumption, $p \in [0,1]$, and also

$$\begin{aligned} p + \mathtt{RL}(\mathcal{H}_{x\to 0}) &= 1 - p + \mathtt{RL}(\mathcal{H}_{x\to 1}) \\ &= \frac{1 + \mathtt{RL}(\mathcal{H}_{x\to 0}) + \mathtt{RL}(\mathcal{H}_{x\to 1})}{2} \\ &\leq \mathtt{RL}(\mathcal{H}). && \text{(Observation 5.5)} \end{aligned}$$

□

```
RandSOA: Randomized SOA

Input: A hypothesis class $\mathcal{H}$.
Initialize: Let $V^{(1)} = \mathcal{H}$.

For $i = 1, 2, \ldots$
    1. Receive $x_i$.
    2. Predict $p_i \in [0,1]$ such that the value
           $\max\left\{p_i + \mathtt{RL}\left(V^{(i)}_{x_i \to 0}\right), 1 - p_i + \mathtt{RL}\left(V^{(i)}_{x_i \to 1}\right)\right\}$        (4)
       is minimized, where $V^{(i)}_{x_i \to b} = \{h \in V^{(i)} : h(x_i) = b\}$.
    3. Receive true label $y_i$.
    4. Update $V^{(i+1)} = V^{(i)}_{x_i \to y_i}$.
```

Figure 3: The randomized SOA is a variation of SOA that finds an optimal randomized prediction in every round. SOA is the name of the original deterministic algorithm by Littlestone [Lit88], and it stands for "Standard Optimal Algorithm".

Lemma 5.7 (Upper bound)**.** *Let $\mathcal{H}$ be a hypothesis class. Then the* RandSOA *learner described in Figure 3 has expected mistake bound*

$$\mathtt{M}(\mathsf{RandSOA}; S) \leq \mathtt{RL}(\mathcal{H})$$

for every realizable input sequence S.

Proof. The proof is by induction on the length of the input sequence. Let $S = (x_1, y_1), \ldots, (x_t, y_t)$ be a realizable sequence. In the base case $t = 0$ we have $\mathtt{M}(\mathsf{RandSOA}; S) = 0 \leq \mathtt{RL}(\mathcal{H})$. For the induction step, assume that $t \geq 1$, and let $S' = (x_2, y_2), \ldots, (x_t, y_t)$ be the input sequence without the first example. In the first round, the learner predicts $p_1 \in [0,1]$ as defined in step 2 of RandSOA. Thus, the learner's expected accumulated loss on S is

$$\mathtt{M}(\mathsf{RandSOA}; S) = |p_1 - y_1| + \mathtt{M}(\mathsf{RandSOA}; S'). \tag{5}$$

By the induction hypothesis we have

$$\mathtt{M}(\mathsf{RandSOA}; S') \leq \mathtt{RL}(\mathcal{H}_{x_1 \to y_1}). \tag{6}$$

Also, by Lemma 5.6 it holds that $p_1 + \mathtt{RL}(\mathcal{H}_{x_1 \to 0}) \leq \mathtt{RL}(\mathcal{H})$ and $1 - p_1 + \mathtt{RL}(\mathcal{H}_{x_1 \to 1}) \leq \mathtt{RL}(\mathcal{H})$, which is equivalent to

$$|p_1 - y_1| + \mathtt{RL}(\mathcal{H}_{x_1 \to y_1}) \leq \mathtt{RL}(\mathcal{H}). \tag{7}$$

Therefore, overall we get that

$$\begin{aligned}
\mathtt{M}(\mathsf{RandSOA}; S) &= |p_1 - y_1| + \mathtt{M}(\mathsf{RandSOA}; S') && \text{(Eq. (5))} \\
&\leq |p_1 - y_1| + \mathtt{RL}(\mathcal{H}_{x_1 \to y_1}) && \text{(Eq. (6))} \\
&\leq \mathtt{RL}(\mathcal{H}), && \text{(Eq. (7))}
\end{aligned}$$

as required. □

5.2 Quasi-balanced Trees

The classical definition of the Littlestone dimension of a class $\mathcal{H}$ is the maximum depth of a balanced (or complete) shattered tree. In contrast, the randomized Littlestone dimension is defined via quantifying over *all* shattered trees. Further, in the deterministic case, balanced trees naturally describe optimal deterministic strategies for the adversary which force any learner to make a mistake on every example along a branch of the tree.

It is therefore natural to ask whether there is a type of shattered trees, analogous to balanced trees, which can be used to define the randomized Littlestone dimension. In this subsection, we show that such an analog exists: a type of trees which we call *quasi-balanced*; roughly speaking, quasi-balanced trees can be seen as a fractional relaxation of balanced trees. We further use this section to prove some useful properties of these trees, which will be used later on.

Informally, quasi-balanced trees are balanced under some weight function defined on the edges. To formally define quasi-balanced trees, we need to define *weight functions* for trees.

Let T be a non-empty tree with edge set E. Let $\mathcal{W} = \mathcal{W}(T)$ be the set of all functions $w\colon E \to [0,1]$, such that for every internal node with outgoing edges e_0, e_1 it holds that $w(e_0) + w(e_1) = 1$. Each function in $\mathcal{W}$ is called a *weight function* for T.

For every branch $b \in B(T)$ defined by a sequence of consecutive edges, define the *weight of the branch b with respect to w* by $w(b) = \sum_{e \in b} w(e)$.

The expected weight of a random branch is always half the expected length of a random branch, as a simple inductive argument shows.

Lemma 5.8. *For every non-empty tree T and every weight function $w \in \mathcal{W}(T)$, the expected weight of a random branch is $E_T/2$.*

Proof. The proof is by induction on the depth of the tree. If T is a leaf then the expected weight of a random branch is $0 = E_T/2$. If T is not a leaf, let e_0, e_1 be the edges emanating from the root, and let T_0, T_1 be the corresponding subtrees. Applying the inductive hypothesis, the expected weight of a random branch in T under w is

$$\frac{w(e_0) + E_{T_0}/2}{2} + \frac{w(e_1) + E_{T_1}/2}{2} = \frac{1 + E_{T_0}/2 + E_{T_1}/2}{2} = E_T/2,$$

using $w(e_0) + w(e_1) = 1$ and Equation (3). □

This lemma prompts the following definition.

Definition 5.9. A tree T is *quasi-balanced* if it is non-empty and there is a weight function $w \in \mathcal{W}(T)$ under which all branches have weight $E_T/2$.

We call $E_T/2$ the *weight* of the tree, and denote it by λ_T.

Lemma 5.10. *If a tree T is quasi-balanced then there is a unique weight function w under which all branches have the same weight. Explicitly, if T' is a subtree of T whose root is connected via edges e_0, e_1 to the subtrees T_0, T_1, then*

$$w(e_0) = \frac{1 + \lambda_{T_1} - \lambda_{T_0}}{2} \text{ and } w(e_1) = \frac{1 + \lambda_{T_0} - \lambda_{T_1}}{2}.$$

Proof. The trees T', T_0, T_1 are necessarily quasi-balanced, and in particular

$$w(e_0) + \lambda_{T_0} = w(e_1) + \lambda_{T_1}.$$

Since $w(e_0) + w(e_1) = 1$, we can solve for $w(e_0), w(e_1)$, obtaining the claimed formula. □

Quasi-balanced trees are a generalization of balanced trees: every tree T which is balanced is also quasi-balanced with weight $\lambda_T = d/2$, where d is the depth of T. This weight is realized by the (unique) constant weight function that gives weight $1/2$ to all edges. The family of quasi-balanced trees is, however, much broader than the family of balanced trees (Figure 4 gives an example of a quasi-balanced tree which is not balanced).

Recall the definition of the randomized Littlestone dimension of the class $\mathcal{H}$:

$$\mathtt{RL}(\mathcal{H}) = \frac{1}{2} \sup_{T \text{ shattered}} E_T.$$

It turns out that in this definition, it suffices to take the supremum only over quasi-balanced trees. This will be easier to see through the characterization of quasi-balanced trees as *monotone* trees.

Definition 5.11 (Monotone Trees)**.** A non-empty tree T is *weakly monotone* if

$$E_T \geq \max\{E_{T_0}, E_{T_1}\},$$

where T_0 and T_1 are the subtrees rooted at the children of the root of T. A tree is *monotone* if it is non-empty and all of its subtrees are weakly monotone.

It is not hard to see that non-monotone trees need not be considered when computing the randomized Littlestone dimension.

Lemma 5.12. *For any non-empty hypothesis class $\mathcal{H}$,*

$$\mathtt{RL}(\mathcal{H}) = \frac{1}{2} \sup_{T \textit{ shattered, monotone}} E_T.$$

Proof. Consider a tree T shattered by $\mathcal{H}$ which is not monotone. Then there exists a vertex v such that $E_{T_v} < E_{T_w}$, where T_w is a tree rooted at a child of v. If we replace the subtree rooted at v with the subtree T_w, we get a tree which is also shattered by $\mathcal{H}$, and has higher expected branch length.

Repeating this process finitely many times, for each tree T shattered by $\mathcal{H}$ we obtain a monotone tree T' shattered by $\mathcal{H}$ satisfying $E_{T'} \geq E_T$, and the lemma follows. □

The following theorem asserts that monotone and quasi-balanced trees are indeed equivalent.

Theorem 5.13. *A tree is quasi-balanced if and only if it is monotone.*

Corollary 5.14. *For any non-empty hypothesis class $\mathcal{H}$,*

$$\mathtt{RL}(\mathcal{H}) = \frac{1}{2} \sup_{T \textit{ shattered, quasi-balanced}} E_T.$$

To prove Theorem 5.13, we use the following simple observation.

Observation 5.15. *Let T be a non-empty tree. Then T is weakly monotone if and only if $|E_{T_0} - E_{T_1}| \leq 2$, where T_0 and T_1 are the subtrees rooted at the children of the root of T.*

Proof. Equation (3) states that $2E_T = 2 + E_{T_0} + E_{T_1}$, and so $E_{T_0} \leq E_T$ is equivalent to $E_{T_0} - E_{T_1} \leq 2$. Similarly, $E_{T_1} \leq E_T$ is equivalent to $E_{T_1} - E_{T_0} \leq 2$. Hence T is weakly monotone iff $|E_{T_0} - E_{T_1}| \leq 2$. □

Proof of Theorem 5.13. An empty tree is neither quasi-balanced nor monotone. Suppose therefore that we are given a non-empty tree T. We prove the equivalence by proving both implications separately.

Monotone $\implies$ Quasi-balanced. The proof is by induction on the depth of the tree. A tree of depth 0 (the base case) is quasi-balanced with weight $E_T/2 = 0$. For the induction step, let T_0, T_1 be the subtrees rooted at the root's children. They are clearly monotone, and so by induction, there are weight functions $w_0 \in \mathcal{W}(T_0)$ and $w_1 \in \mathcal{W}(T_1)$ under which all branches in T_0 have weight $\lambda_{T_0} = E_{T_0}/2$ and all branches in T_1 have weight $\lambda_{T_1} = E_{T_1}/2$.

Let e_0, e_1 be the edges connecting the root of T to the roots of T_0, T_1, respectively. Define a weight function $w \in \mathcal{W}(T)$ by defining $w(e) = w_0(e)$ if $e \in T_0$, $w(e) = w_1(e)$ if $e \in T_1$,

$$w(e_0) = \frac{1 + \lambda_{T_1} - \lambda_{T_0}}{2}, \quad \text{and} \quad w(e_1) = \frac{1 + \lambda_{T_0} - \lambda_{T_1}}{2}.$$

Clearly $w(e_0) + w(e_1) = 1$. Observation 5.15 implies that $w(e_0), w(e_1) \in [0, 1]$, and so indeed $w \in \mathcal{W}(T)$. Since $w(e_0)+\lambda_{T_0} = w(e_1)+\lambda_{T_1}$, the weight function w shows that T is quasi-balanced.

Quasi-balanced $\implies$ Monotone. The proof is by induction on the depth of the tree. A tree of depth 0 is monotone. For the induction step, we first observe that every proper subtree of T is quasi-balanced, and so monotone by the inductive hypothesis. Hence it suffices to show that T is weakly monotone.

Let w be the unique weight function for T under which each branch has weight λ_T. Let e_0, e_1 be the edges connecting the root of T to the two subtrees T_0, T_1. According to Lemma 5.10, the weights of these edges are

$$w(e_0) = \frac{1 + \lambda_{T_1} - \lambda_{T_0}}{2} \text{ and } w(e_1) = \frac{1 + \lambda_{T_0} - \lambda_{T_1}}{2}.$$

Since the weights are non-negative, we deduce that $|\lambda_{T_0} - \lambda_{T_1}| \leq 1$, and so $|E_{T_0} - E_{T_1}| \leq 2$. We conclude that T is weakly monotone by Observation 5.15. □

5.3 Applications of Quasi-Balanced Trees

We now give two applications of quasi-balanced trees. In Section 5.3.1 we show that they can be used to give explicit strategies for the adversary. In Section 5.3.2 we provide an alternative proof for the folklore inequality $\mathtt{M}^\star(\mathcal{H}) \leq \mathtt{M}^\star_D(\mathcal{H}) \leq 2\mathtt{M}^\star(\mathcal{H})$, which appears implicitly in [BDPSS09].

5.3.1 Optimal Online Adversarial Strategies

Lemma 5.3 states that for every learning rule Lrn there exists a realizable sequence S so that $\mathtt{M}(\mathsf{Lrn}; S) \geq E_T/2$. In the proof we showed that if S is chosen according to a random branch, then $\mathbb{E}[\mathtt{M}(\mathsf{Lrn}; S)] \geq E_T/2$.

Quasi-balanced trees allow us to explicitly describe strategies which approach $E_T/2$.

Lemma 5.16. *Let $\mathcal{H}$ be a non-empty hypothesis class, and let T be a quasi-balanced tree shattered by $\mathcal{H}$, as witnessed by $w \in \mathcal{W}(T)$. Let Lrn be an arbitrary learning rule. Consider the following strategy for the adversary, which traverses T from the root to a leaf, and acts as follows at step i, when at a node v_i with outgoing edges e_0, e_1:*

1. *Send the learner the label x_i of v_i, receiving the answer $p_i \in [0, 1]$.*

2. *If $p_i \geq w(e_0)$ then set the true label to 0 and proceed accordingly.*

3. *Otherwise set the true label to 1 and proceed accordingly.*

Then the resulting sequence S of examples is realizable by $\mathcal{H}$ and satisfies $\mathtt{M}(\mathsf{Lrn}; S) \geq E_T/2$.

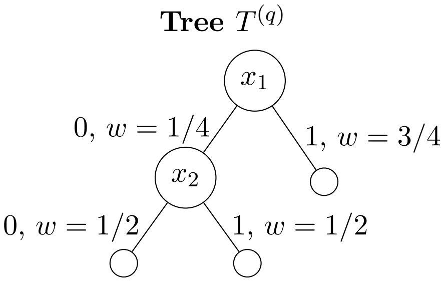

Figure 4: The tree $T^{(q)}$ is a quasi-balanced tree with weight $\lambda_{T^{(q)}} = 3/4 = E_{T^{(q)}}/2$, which is realized by the weight function w written on the edges. The internal nodes are associated with instances x_1, x_2. The function w guides the adversary's strategy: Determine x_1 to be the instance in the first round. If $p_1 \leq 1/4$, determine $y_1 = 1$ and finish the game. Otherwise, set $y_1 = 0$, determine the instance in the second round to be x_2, and finish the game after the second round. Either way, the learner will suffer a loss of at least $3/4$ in total.

Proof. It is clear that S is realizable. If $p_i \geq w(e_0)$ then the loss incurred by the learner at step i is $|p_i - 0| \geq w(e_0)$. Otherwise, it is $|1 - p_i| \geq |1 - w(e_0)| = w(e_1)$. Since every path in T has weight exactly $E_T/2$, it follows that the loss of the learner is at least $E_T/2$. □

An example can be found in Figure 4.

5.3.2 Deterministic vs Randomized Online Learning

Quasi-balanced trees can be used to give an alternative proof for the following well-known relation between the randomized and deterministic mistake bounds.

Proposition 5.17 ([BDPSS09]). *Let $\mathcal{H} \neq \emptyset$ be a hypothesis class. Then*

$$\mathtt{M}^\star(\mathcal{H}) \leq \mathtt{M}^\star_D(\mathcal{H}) \leq 2\mathtt{M}^\star(\mathcal{H}).$$

Classic proof. It is obvious that $\mathtt{M}^\star(\mathcal{H}) \leq \mathtt{M}^\star_D(\mathcal{H})$, because a deterministic learner is also a special case of a randomized learner.

The inequality $\mathtt{M}^\star(\mathcal{H}) \leq 2\mathtt{M}^\star_D(\mathcal{H})$ follows by a simple derandomization which transforms any randomized learner Lrn to a deterministic learner Lrn_D whose mistake bound is at most twice as large. Specifically, Lrn_D is defined as follows. Let S be an input sequence of examples, and let p_i denote the prediction of Lrn on the i'th example in S. Lrn_D predicts 0 if $p_i \leq 1/2$ and 1 otherwise. Notice that whenever Lrn_D makes a mistake, the loss of Lrn increases by at least $1/2$. Thus, the total number of mistakes made by Lrn_D is at most twice the loss of Lrn. □

Using Theorem 5.2 we can give an alternative proof of Proposition 5.17, which uses the original characterization for the deterministic setting from [Lit88]. Specifically, we can formulate Proposition 5.17 in terms of the Littlestone and randomized Littlestone dimensions, and prove it directly using properties of quasi-balanced trees.

The heart of the proof is the following simple lemma, showing that the expected branch length of a quasi-balanced tree is at most twice the minimum branch length.

Proposition 5.18. *If T is a quasi-balanced tree then $E_T \leq 2m_T$, where m_T is the minimum length of a branch of T.*

The tree $T^{(nq)}$

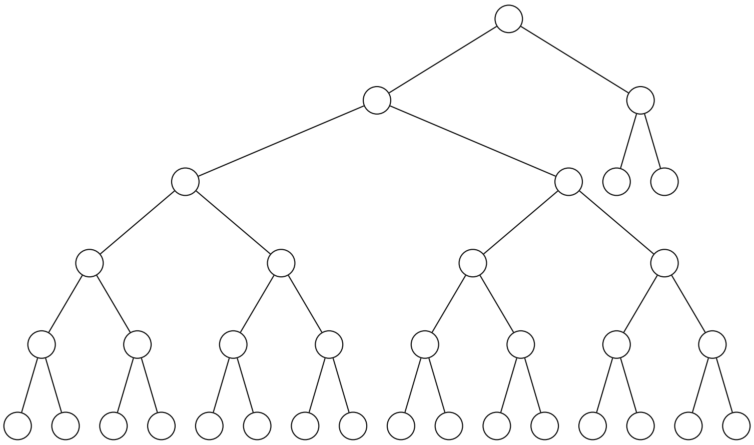

Figure 5: The minimal branch in $T^{(nq)}$ is of length 2, while $E_{T^{(nq)}} = 3.5$. Therefore it holds that $E_{T^{(nq)}}$ is at most twice the minimal branch length. Since every proper subtree of $T^{(nq)}$ is complete, this also holds for all proper subtrees. Nevertheless, $T^{(nq)}$ is not quasi-balanced, since it is not monotone.

Proof. The proof is by induction on the depth of T. If T consists of a single vertex then $E_T = m_T = 0$. Otherwise, let T_0, T_1 be the subtrees rooted at the children of the root of T. Applying Equation (3), we get

$$E_T = 1 + E_{T_0}/2 + E_{T_1}/2 = 1 + \min(E_{T_0}, E_{T_1}) + |E_{T_0} - E_{T_1}|/2.$$

Observation 5.15 shows that $|E_{T_0} - E_{T_1}|/2 \leq 1$, and so applying the inductive hypothesis, we see that

$$E_T \leq 2 + 2\min(m_{T_0}, m_{T_1}) = 2m_T. \qquad \square$$

We can now give the alternative proof of Proposition 5.17.

Alternative proof of Proposition 5.17. Since $\mathtt{M}^\star(\mathcal{H}) = \mathtt{RL}(\mathcal{H})$ by Theorem 5.2 and $\mathtt{M}^\star_D(\mathcal{H}) = \mathtt{L}(\mathcal{H})$ by Theorem 4.1, it suffices to prove that

$$\mathtt{RL}(\mathcal{H}) \leq \mathtt{L}(\mathcal{H}) \leq 2\mathtt{RL}(\mathcal{H}).$$

The inequality $\mathtt{L}(\mathcal{H}) \leq 2\mathtt{RL}(\mathcal{H})$ easily follows from the definitions:[12]

$$\mathtt{L}(\mathcal{H}) = \sup_{T \text{ shattered}} m_T \quad \text{and} \quad \mathtt{RL}(\mathcal{H}) = \sup_{T \text{ shattered}} E_T.$$

Indeed, the expected depth of a random branch is always at least the minimum depth of a branch.

In order to prove the inequality $\mathtt{RL}(\mathcal{H}) \leq \mathtt{L}(\mathcal{H})$, we use Corollary 5.14, which allows us to restrict the trees in the definition of $\mathtt{RL}(\mathcal{H})$ to be quasi-balanced. The inequality then immediately follows from Proposition 5.18. □

Unlike Theorem 5.13, the property of quasi-balanced trees proved in Proposition 5.18 is not a characterization of quasi-balanced trees. Figure 5 gives an example for a tree that satisfies this property but is not quasi-balanced.

Both inequalities in Proposition 5.17 can be tight, as the following examples demonstrate.

[12] Notice that in the classic proof, the other inequality is the trivial one.

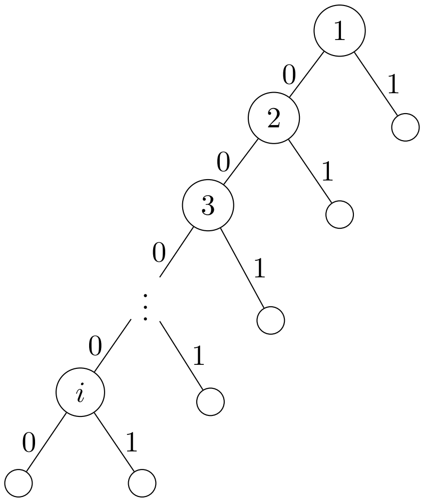

Figure 6: The tree T_i defined in Example 5.19.

Example 5.19 (class $\mathcal{H}_1$ with $\mathtt{RL}(\mathcal{H}_1) = \mathtt{L}(\mathcal{H}_1)$)**.** *Let $\mathcal{H}_1$ be the class of singletons over $\mathbb{N}$. That is, $\mathcal{X} = \mathbb{N}$ and $\mathcal{H}_1 = \{h_i : i \in \mathbb{N}\}$, where $h_i(j) = 1$ if and only if $i = j$. Any tree shattered by $\mathcal{H}_1$ has minimum branch length* 1 *(since no hypothesis satisfies $h(i) = h(j) = 1$ for $i \neq j$), hence $\mathtt{L}(\mathcal{H}_1) = 1$. In contrast, the tree T_i in Figure 6 is shattered by $\mathcal{H}_1$ and has expected branch length $2 - 2^{-i}$, and so $\mathtt{RL}(\mathcal{H}_1) \geq 1$.*

In Section 5.4 we show how to extend the definition of randomized Littlestone dimension to infinite *trees. We can then replace the trees T_i with a single infinite tree T_∞ shattered by $\mathcal{H}_1$ whose expected branch length is exactly* 2*.*

Example 5.20 (Class $\mathcal{H}_2$ with $\mathtt{RL}(\mathcal{H}_2) = \mathtt{L}(\mathcal{H}_2)/2$)**.** *Let $\mathcal{X} = \{1\}$ and let $\mathcal{H}_2 = \{h_0, h_1\}$, where $h_\ell(1) = \ell$. There are only two non-empty trees shattered by $\mathcal{H}_2$: a leaf, and the complete binary tree of depth* 1 *whose root is labelled* 1*. Hence $\mathtt{L}(\mathcal{H}_2) = 1$ and $\mathtt{RL}(\mathcal{H}_2) = 1/2$.*

5.4 Infinite Trees

So far we have been considering only finite trees. However, in the sequel it will be useful to also allow infinite trees. In this short subsection, we extend the definition of E_T to infinite trees, and show that the formula for $\mathtt{RL}$ holds even when allowing infinite trees.

In this section, whenever we refer to trees, we mean full ordered binary trees, which are possibly infinite. A tree is *shattered* by a hypothesis class $\mathcal{H}$ if every (possibly infinite) path starting at the root is realizable by $\mathcal{H}$.

We define a *random path* in the same way that we defined a random branch in the finite case: start at the root, and at each internal vertex, choose a random child at uniform, stopping if a leaf is reached. The result is either a (finite) branch or an infinite path.

We define E_T as the expected length of a random path. If the random path is finite almost surely, then E_T is given using the same formula as in the finite case:

$$E_T = \sum_{b \in B(T)} |b| \cdot 2^{-|b|}.$$

Figure 7 gives an example of such a tree. If the random path is infinite with positive probability, then $E_T = \infty$. It can also happen that $E_T = \infty$ if the random path is finite almost surely.

The formula in Definition 5.1 holds even if we allow infinite trees.

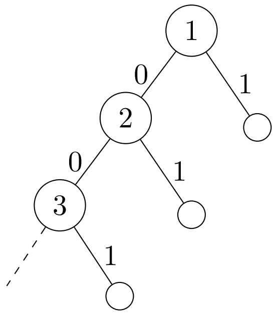

Figure 7: An infinite tree with finite expected branch length 2.

Lemma 5.21. *For any non-empty hypothesis class $\mathcal{H}$,*

$$\mathtt{RL}(\mathcal{H}) = \frac{1}{2} \sup_{T \textit{ shattered}} E_T,$$

where the supremum is taken over possibly infinite *trees.*

The proof of the lemma uses a straightforward truncation argument.

Proof. Substituting the definition of $\mathtt{RL}(\mathcal{H})$, we need to prove that

$$\frac{1}{2} \sup_{T \text{ shattered, finite}} E_T = \frac{1}{2} \sup_{T \text{ shattered}} E_T.$$

The left-hand side is clearly at most the right-hand side. We show that they coincide by constructing, for each infinite shattered tree T, a sequence of finite shattered trees T_D such that

$$E_T = \lim_{D \to \infty} E_{T_D}. \tag{8}$$

The tree T_D is simply the truncation of T to depth D (that is, all branches of T_D have length at most D). To prove Equation (8), let $\Lambda \in \mathbb{N} \cup \{\infty\}$ be the length of a random path in T, and let $\Lambda_D \in \{0, \dots, D\}$ be the length of a random path in T_D. We can couple the random paths so that $\Lambda_D = \min(\Lambda, D)$.

If Λ is almost surely finite then

$$E_T = \sum_{\ell \in \mathbb{N}} \ell \Pr[\Lambda = \ell].$$

Equation (8) holds because on the one hand, $E_{T_D} \le E_T$, and on the other hand,

$$E_{T_D} \ge \sum_{\ell \le D} \ell \Pr[\Lambda = \ell] \xrightarrow{D \to \infty} E_T.$$

In contrast, if $p := \Pr[\Lambda = \infty] > 0$ then $\Pr[\Lambda_D = D] \ge p$ and so $E_{T_D} \ge pD \to \infty$, hence Equation (8) again holds. □

5.5 Trees Maximizing the Expected Branch Length

The sequence of trees $(T_i)_{i=0}^{\infty}$ described in Example 5.19 suggests that for "well-behaved classes", the supremum in Theorem 5.2 is attained by a specific tree. We show that this is true for finite classes.

Proposition 5.22. *Let $\mathcal{H}$ be a* finite *hypothesis class. Then there exists a tree shattered by $\mathcal{H}$ such that*

$$\mathtt{RL}(\mathcal{H}) = \frac{1}{2} E_T.$$

Proof. Let T be a tree shattered by $\mathcal{H}$. We can label each branch of T by a hypothesis realizing it. Each branch must be labeled by a different hypothesis, hence the number of branches is at most $|\mathcal{H}|$. Consequently, there are only finitely many shattered trees, and so the supremum in the definition of $\mathtt{RL}(\mathcal{H})$ is trivially achieved. □

There are also classes for which the maximum is not attained, even if we allow infinite trees.

Example 5.23 (Maximum is not necessarily attained for infinite classes)**.** *We construct a hypothesis class $\mathcal{H}$ over the domain $\mathcal{X} = \{(i,j) \in \mathbb{N}^2 : 1 \le i \le j\}$. For each $(i,j) \in \mathcal{X}$, the hypothesis class $\mathcal{H}$ contains the function*

$$h_{i,j}(I,J) = \begin{cases} 1 & \text{if } J \neq j, \\ 1[i = I] & \text{if } J = j. \end{cases}$$

Let us start by computing $\mathtt{L}(\mathcal{H})$. Consider any tree T shattered by $\mathcal{H}$ which is not a leaf. Suppose that the root is labelled by (i,j). Let T_0 be the subtree rooted at the branch of the root labelled 0. Since no hypothesis in $\mathcal{H}$ gives 0 to inputs with different second parts, all vertices in T_0 must be labelled by (i',j). Since no hypothesis in $\mathcal{H}$ gives 0 to (i,j) and 1 to two different (i',j), we see that the minimum branch length in T_0 is at most 1, and so the minimum branch length in T is at most 2. Hence $\mathtt{L}(\mathcal{H}) \le 2$. It is easy to construct a tree showing that $\mathtt{L}(\mathcal{H}) = 2$.

The subtree T_0 contains at most j branches, and each edge labelled 1 terminates at a leaf. A simple induction on j shows that the expected branch length of such a tree is strictly less than 2. Indeed, denoting the expected branch length for a given value of j by A_j, we have $A_1 = 0$ and $A_j = (A_{j-1} + 1)/2$, and so $A_j = 2(1 - 2^{1-j})$.

Let j_s be the number of leaves in the subtree rooted at the vertex obtained by starting at the root of T, taking s times the outgoing edge labelled 1, and then the outgoing edge labelled 0 (if such a vertex exists). Thus

$$E_T = \sum_{s=0}^{S} 2^{-s-1}(s + 1 + A_{j_s}) < \sum_{s=0}^{S} 2^{-s-1}(s+3) = 4,$$

where S is the maximal value for which j_s is defined (possibly $S = \infty$).

On the other hand, we can construct a tree T shattered by $\mathcal{H}$ for which E_T is arbitrarily close to 4. Start with an infinite right path (that is, a path in which all edges are labelled 1) labelled with $(1,K), (1,K+1), (1,K+2)$ and so on, for some parameter K. The left branch of a vertex labelled $(1,J)$ is labelled using $(2,J), \ldots, (J-1,J)$ to construct a tree T'_J shattered by $\mathcal{H}$ with $J-1$ branches, as described in Figure 8. This tree satisfies $j_s = K + s - 1$, and so

$$E_T = \sum_{s=0}^{\infty} 2^{-s-1}(s + 3 - 2^{1-K-s}) = 4 - O(2^{-K}),$$

which is arbitrarily close to 4. Thus $\mathtt{RL}(\mathcal{H}) = 2$, but every (possibly infinite) tree T shattered by $\mathcal{H}$ satisfies $E_T/2 < 2$.

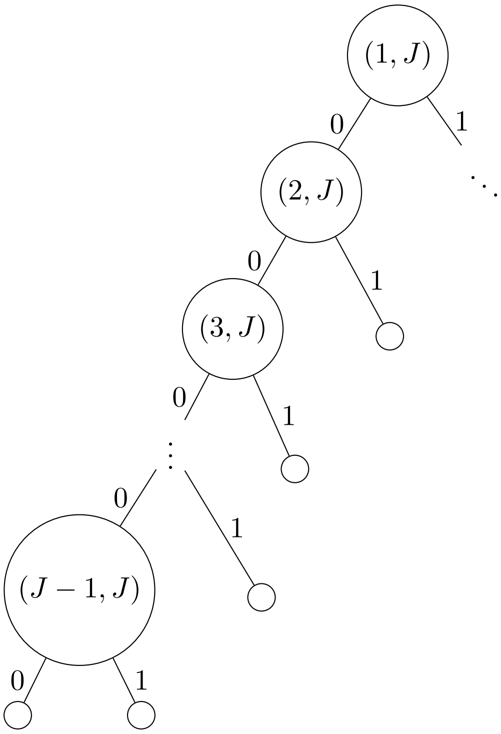

Figure 8: The tree T'_J defined in Example 5.23.

6 Bounded Horizon

So far we have not put any restrictions on the number of rounds. However, in many circumstances we are interested in the online learning game when the number of rounds is bounded. We model this by assuming that the learner knows an upper bound on the number of rounds. We define $\mathtt{M}^\star(\mathcal{H}, \mathbf{T})$ to be the optimal randomized mistake bound when the number of rounds is at most $\mathbf{T}$.

We can generalize Theorem 5.2 to this setting. The required notion of randomized Littlestone dimension is

$$\mathtt{RL}(\mathcal{H}, \mathbf{T}) = \frac{1}{2} \sup_{\substack{T \text{ shattered} \\ \mathsf{depth}(T) \leq \mathbf{T}}} E_T.$$

The bounded randomized Littlestone dimension gives the precise mistake bound in this setting.

Theorem 6.1 (Optimal Randomized Mistake Bound with Finite Horizon)**.** *Let $\mathcal{H}$ be a hypothesis class, and let $\mathbf{T} \in \mathbb{N}$. Then,*

$$\mathtt{M}^\star(\mathcal{H}, \mathbf{T}) = \mathtt{RL}(\mathcal{H}, \mathbf{T}).$$

We prove Theorem 6.1 in Section 6.1. This theorem immediately suggests the following questions:

1. How many rounds are needed in order for the adversary to guarantee that the loss of the learner is at least $\mathtt{RL}(\mathcal{H}) - \epsilon$?
   We prove in Section 6.3 that $2\mathtt{RL}(\mathcal{H}) + O(\log(1/\epsilon))$ rounds suffice, and discuss the optimality of this bound in Section 6.5.

2. What can we say about the loss of the learner when there are fewer than $2\mathtt{RL}(\mathcal{H})$ rounds? A trivial upper bound on $\mathtt{RL}(\mathcal{H}, \mathbf{T})$ is $\mathbf{T}/2$. In Section 6.4 we show that this bound is nearly optimal when $\mathbf{T} \leq 2\mathtt{RL}(\mathcal{H})$.

```
BoundedRandSOA: Bounded Randomized SOA

Input: A hypothesis class $\mathcal{H}$ and number of rounds $\mathbf{T}$.
Initialize: Let $V^{(1)} = \mathcal{H}$.

For $i = 1, 2, \dots, \mathbf{T}$
    1. Receive $x_i$.
    2. Predict $p_i \in [0,1]$ such that the value
           $\max\Big\{p_i + \mathtt{RL}\Big(V^{(i)}_{x_i \to 0}, \mathbf{T} - i\Big), 1 - p_i + \mathtt{RL}\Big(V^{(i)}_{x_i \to 1}, \mathbf{T} - i\Big)\Big\}$
       is minimized, where $V^{(i)}_{x_i \to b} = \{h \in V^{(i)} : h(x_i) = b\}$.
    3. Receive true label $y_i$.
    4. Update $V^{(i+1)} = V^{(i)}_{x_i \to y_i}$.
```

Figure 9: BoundedRandSOA is a bounded variant of RandSOA.

The proofs of these results use concentration bounds on the depth of quasi-balanced trees, which we prove in Section 6.2.

6.1 Proof of Theorem 6.1

In this section we indicate how to generalize the proof of Theorem 5.2 to the finite horizon setting, proving Theorem 6.1.

The lower bound $\mathtt{RL}(\mathcal{H}, \mathbf{T}) \le \mathtt{M}^\star(\mathcal{H}, \mathbf{T})$ follows directly from the statement of Lemma 5.3, since the length of S is at most $\mathsf{depth}(T)$.

For the upper bound, we use a straightforward modification of algorithm RandSOA, which appears in Figure 9.

We start by extending Observation 5.5: if $\mathcal{H}$ is a non-empty hypothesis class and $\mathbf{T} > 0$ then

$$\mathtt{RL}(\mathcal{H}, \mathbf{T}) = \frac{1}{2} \max_{x \in \mathcal{X}} \big(1 + \mathtt{RL}(\mathcal{H}_{x \to 0}, \mathbf{T} - 1) + \mathtt{RL}(\mathcal{H}_{x \to 1}, \mathbf{T} - 1)\big).$$

The proof is identical. Since there are only finitely many unlabelled trees of depth at most $\mathbf{T}$, we can replace the supremum with a maximum.

The next step is to generalize Lemma 5.6, which now states that for every hypothesis class $\mathcal{H}$, instance $x \in \mathcal{X}$, and $\mathbf{T} > 0$, there exists $p \in [0,1]$ so that

$$p + \mathtt{RL}(\mathcal{H}_{x \to 0}, \mathbf{T} - 1) \le \mathtt{RL}(\mathcal{H}, \mathbf{T}) \quad \text{and} \quad (1 - p) + \mathtt{RL}(\mathcal{H}_{x \to 1}, \mathbf{T} - 1) \le \mathtt{RL}(\mathcal{H}, \mathbf{T}).$$

The proof is identical, using the generalized Observation 5.5.

Finally, we prove the following generalization of Lemma 5.7: for every hypothesis class $\mathcal{H}$, any parameter $\mathbf{T}$, and any realizable input sequence S of length at most $\mathbf{T}$,

$$\mathtt{M}(\mathsf{BoundedRandSOA}; S) \le \mathtt{RL}(\mathcal{H}, \mathbf{T}).$$

The proof is identical, using the generalized Lemma 5.6.

6.2 A Concentration Lemma for Quasi-Balanced Trees

Another interesting property of quasi-balanced trees is that the length of a random branch concentrates around its expectation. This property will be important for deriving tight bounds in Section 8.

Proposition 6.2 (Concentration of branch lengths)**.** *Let T be a quasi-balanced tree, and let X be the length of a random branch. Then for any $\epsilon > 0$,*

$$\Pr[X < (1-\epsilon)E_T] \le \exp(-\epsilon^2 E_T/4) \quad \textit{and} \quad \Pr[X > (1+\epsilon)E_T] \le \exp(-\epsilon^2 E_T/4(1+\epsilon)).$$

Proof. If T is a single leaf then the result trivially holds since there is a single random branch. Therefore we can assume that T is not a single leaf, and in particular, $E_T \ge 1$.

Let $b_0, b_1, b_2, \dots$ be an infinite sequence of random coin tosses. We can choose a random branch of T as follows. Let v_0 be the root of T. For $i \in \mathbb{N}$, if v_i is not a leaf, then v_{i+1} is obtained by following the edge labelled b_i. Otherwise, we define $v_{i+1} = v_i$. The resulting random branch has exactly the same distribution that we have been considering so far.

Let L_i be the expected length of the branch given $b_0, \dots, b_{i-1}$. This is an exposure martingale, as defined in Section 4.

In order to apply Azuma's inequality, we need to bound the random difference $|L_i - L_{i+1}|$. If v_i is a leaf, then $L_{i+1} = L_i$. Otherwise, let T' be the subtree rooted at v_i, and let T'_0, T'_1 be the subtrees rooted at the children of v_i. Thus L_{i+1} is either $\lambda_0 := i + 1 + E_{T'_0}$ or $\lambda_1 := i + 1 + E_{T'_1}$, depending on the value of b_i. Moreover, $L_i = (\lambda_0 + \lambda_1)/2$ is the average of these two values.

Theorem 5.13 shows that T' is weakly monotone, and so Observation 5.15 shows that $|E_{T'_0} - E_{T'_1}| \le 2$. Consequently,

$$|L_i - L_{i+1}| = \frac{1}{2}|\lambda_0 - \lambda_1| \le 1.$$

The definition of L_i implies that $L_\beta = X$ for all $\beta \ge X$. In particular, if $X < (1-\epsilon)E_T$ then $L_{\lceil E_T \rceil} < (1-\epsilon)E_T$. Applying Azuma's inequality and using $L_0 = E_T$, it follows that

$$\Pr[X < (1-\epsilon)E_T] \le \Pr[L_{\lceil E_T \rceil} - E_T < -\epsilon E_T] \le \exp\left(\frac{-\epsilon^2 E_T^2}{2\lceil E_T \rceil}\right) \le \exp(-\epsilon^2 E_T/4),$$

where the final inequality uses $\lceil E_T \rceil \le E_T + 1 \le 2E_T$.

The definition of L_i also implies that $L_\beta \ge \beta$ whenever $\beta \le X$. In particular, if $X > (1+\epsilon)E_T$ then $L_{\lceil (1+\epsilon)E_T \rceil} \ge \lceil (1+\epsilon)E_T \rceil$. Therefore

$$\Pr[X > (1+\epsilon)E_T] \le \Pr[L_{\lceil (1+\epsilon)E_T \rceil} - E_T > \epsilon E_T] \le \exp\left(\frac{-\epsilon^2 E_T^2}{2\lceil (1+\epsilon)E_T \rceil}\right) \le \exp(-\epsilon^2 E_T/4(1+\epsilon)),$$

using $(1+\epsilon)E_T \ge 1$ as before. □

6.3 Approaching $\mathtt{RL}(\mathcal{H})$

As a simple consequence of the concentration bound proved in Proposition 6.2, we show that we can approach $\mathtt{RL}(\mathcal{H})$ using relatively shallow trees, quantified as follows.

Proposition 6.3. *Let $\mathcal{H}$ be a non-empty hypothesis class with finite randomized Littlestone dimension $\mathtt{RL}(\mathcal{H})$.*

For every $\epsilon > 0$ there is a tree T shattered by $\mathcal{H}$ satisfying $E_T/2 \ge \mathtt{RL}(\mathcal{H}) - \epsilon$ whose depth is at most $2\mathtt{RL}(\mathcal{H}) + O(\log(1/\epsilon))$.

Given Lemma 5.16, this means that the adversary can force the learner to suffer a loss of $\mathtt{RL}(\mathcal{H}) - \epsilon$ after only $2\mathtt{RL}(\mathcal{H}) + O(\log(1/\epsilon))$ rounds. In contrast, at least $2\mathtt{RL}(\mathcal{H}) - 2\epsilon$ rounds are clearly needed, since a learner who predicts $1/2$ at each round suffers a loss of $R/2$ after R rounds.

We prove Proposition 6.3 via the following technical estimate.

Lemma 6.4. *Let T be a monotone tree, and let $T^{\leq k}$ result from truncating it to the first k levels (all branches in $T^{\leq k}$ have length at most k). If $k \geq 2E_T$ then*

$$E_{T^{\leq k}} \geq E_T - 8e^{(E_T - k)/8}.$$

Proof. Let X be the length of a random branch of T. Using X, we can express the difference between $E_{T^{\leq k}}$ and E_T explicitly:

$$E_T - E_{T^{\leq k}} = \sum_{t>k} \Pr[X \geq t].$$

Applying Proposition 6.2, we can bound each term by

$$\exp(-\epsilon^2 E_T / 4(1+\epsilon)),$$

where $\epsilon = t/E_T - 1$. By assumption, $\epsilon \geq 1$, and so

$$\frac{\epsilon^2 E_T}{4(1+\epsilon)} \geq \frac{\epsilon E_T}{8} = \frac{t - E_T}{8}.$$

Therefore

$$E_T - E_{T^{\leq k}} \leq \sum_{t>k} e^{(E_T - t)/8} \leq e^{E_T/8} \int_k^\infty e^{-t/8}\, dt = 8e^{(E_T-k)/8}. \qquad \square$$

We can now prove Proposition 6.3.

Proof of Proposition 6.3. Let $k = 2\mathtt{RL}(\mathcal{H}) + 8\ln(8/\epsilon)$. Applying Lemma 5.12, we can find a monotone tree T shattered by $\mathcal{H}$ such that $\mathtt{RL}(\mathcal{H}) - \epsilon/2 \leq E_T/2 \leq \mathtt{RL}(\mathcal{H})$. Lemma 6.4 implies that $E_{T^{\leq k}} \geq E_T - \epsilon$, and so $E_{T^{\leq k}}/2 \geq \mathtt{RL}(\mathcal{H}) - \epsilon$. □

We discuss the optimality of Proposition 6.3 in Section 6.5.

6.4 Mistake Bound for Few Rounds

Another truncation argument allows us to estimate $\mathtt{RL}(\mathcal{H}, \mathbf{T})$ for small $\mathbf{T}$.

Proposition 6.5. *Let $\mathcal{H}$ be a non-empty hypothesis class with finite randomized Littlestone dimension $\mathtt{RL}(\mathcal{H})$.*

If $\mathbf{T} \leq \mathtt{RL}(\mathcal{H})$ then $\mathtt{RL}(\mathcal{H}, \mathbf{T}) = \mathbf{T}/2$.

If $\mathbf{T} \leq 2\mathtt{RL}(\mathcal{H})$ then

$$\frac{\mathbf{T}}{2} - O(\sqrt{\mathbf{T} \log \mathbf{T}}) \leq \mathtt{RL}(\mathcal{H}, \mathbf{T}) \leq \frac{\mathbf{T}}{2}.$$

Furthermore, if $\mathbf{T} \leq 2\mathtt{RL}(\mathcal{H}) - \sqrt{8\mathtt{RL}(\mathcal{H}) \ln \mathtt{RL}(\mathcal{H})}$ then

$$\frac{\mathbf{T}}{2} - 1 < \mathtt{RL}(\mathcal{H}, \mathbf{T}) \leq \frac{\mathbf{T}}{2}.$$

Proof. A learner that always predicts $1/2$ suffers a loss of exactly $1/2$ each round, showing that $\mathtt{RL}(\mathcal{H},\mathbf{T}) \le \mathbf{T}/2$ for each $\mathbf{T}$. In contrast, if T is a tree shattered by $\mathcal{H}$ then Theorem 6.1 shows that $\mathtt{RL}(\mathcal{H},\mathbf{T}) \ge E_{T^{\le \mathbf{T}}}/2$, and we will use this to give lower bounds on $\mathtt{RL}(\mathcal{H},\mathbf{T})$.

Suppose first that $\mathbf{T} \le \mathtt{RL}(\mathcal{H})$. Proposition 5.18 shows that $m_T \ge E_T/2$. If E_T is close enough to $2\mathtt{RL}(\mathcal{H})$ then $m_T \ge \mathtt{RL}(\mathcal{H})$ (since m_T is an integer), and so $T^{\le \mathbf{T}}$ is a complete tree of depth $\mathbf{T}$. This shows that $\mathtt{RL}(\mathcal{H},\mathbf{T}) \ge \mathbf{T}/2$.

In order to prove the remaining results, suppose that $\mathbf{T} \le 2\mathtt{RL}(\mathcal{H})$, and consider a tree T shattered by $\mathcal{H}$ satisfying $E_T = 2\mathtt{RL}(\mathcal{H}) - \delta \ge \mathbf{T}$. Proposition 6.2 shows that a random branch of $T^{\le \mathbf{T}}$ has depth $\mathbf{T}$ with probability at least $1 - \exp\bigl(-\frac{(E_T-\mathbf{T})^2}{4E_T}\bigr)$, and so

$$\mathtt{RL}(\mathcal{H},\mathbf{T}) \ge \left(1 - \exp\left(-\frac{(E_T-\mathbf{T})^2}{4E_T}\right)\right)\cdot\frac{\mathbf{T}}{2} \longrightarrow \left(1 - \exp\left(-\frac{(2\mathtt{RL}(\mathcal{H})-\mathbf{T})^2}{8\mathtt{RL}(\mathcal{H})}\right)\right)\cdot\frac{\mathbf{T}}{2},$$

where the limit is taken along a sequence of trees shattered by $\mathcal{H}$ and satisfying $E_T \to 2\mathtt{RL}(\mathcal{H})$.

If $\mathbf{T} \le \mathbf{T}_0 := 2\mathtt{RL}(\mathcal{H}) - \sqrt{8\mathtt{RL}(\mathcal{H})\ln \mathtt{RL}(\mathcal{H})}$ then this gives

$$\mathtt{RL}(\mathcal{H},\mathbf{T}) \ge \left(1 - \frac{1}{\mathtt{RL}(\mathcal{H})}\right)\cdot\frac{\mathbf{T}}{2} > \frac{\mathbf{T}}{2} - 1.$$

If $\mathbf{T}_0 \le \mathbf{T} \le 2\mathtt{RL}(\mathcal{H})$ then

$$\mathtt{RL}(\mathcal{H},\mathbf{T}) \ge \mathtt{RL}(\mathcal{H},\mathbf{T}_0) \ge \frac{\mathbf{T}_0-2}{2} \ge \frac{\mathbf{T}-2}{2} - \sqrt{8\mathtt{RL}(\mathcal{H})\ln \mathtt{RL}(\mathcal{H})} \ge \frac{\mathbf{T}}{2} - O(\sqrt{\mathbf{T}\log\mathbf{T}}). \qquad \square$$

6.5 Lower Bounds

Let $\mathcal{H}$ by a hypothesis class. If there exists a (finite) tree T shattered by $\mathcal{H}$ and satisfying $E_T/2 = \mathtt{RL}(\mathcal{H})$, then Proposition 6.3 is not tight for small $\epsilon > 0$. Proposition 5.22 shows that such a tree always exists when $\mathcal{H}$ is finite. Conversely, when $\mathcal{H}$ is infinite, we can show that the additive factor $O(\log(1/\epsilon))$ is necessary.

We start by showing that $\mathtt{RL}(\mathcal{H}) \ge 1$ if $\mathcal{H}$ is infinite.

Lemma 6.6. *Let $\mathcal{H}$ be a hypothesis class. If $|\mathcal{H}| \ge k$ then there is a tree T shattered by $\mathcal{H}$ such that $E_T \ge 2 - 2^{2-k}$. In particular, if $\mathcal{H}$ is infinite then $\mathtt{RL}(\mathcal{H}) \ge 1$.*

Proof. The proof is by induction on k. If $k = 1$ then there is nothing to prove. Otherwise, $|\mathcal{H}| \ge 2$, and so there exists an instance x such that $\mathcal{H}_{x\to 0}, \mathcal{H}_{x\to 1} \ne \emptyset$. If $|\mathcal{H}_{x\to y}| = k_y$, then using the induction hypothesis, we construct a tree T shattered by $\mathcal{H}$ such that

$$E_T \ge 1 + \frac{2-2^{2-k_0}}{2} + \frac{2-2^{2-k_1}}{2}.$$

By convexity, the right-hand side is minimized when $k_0 = 1$ and $k_1 = k-1$, and so $E_T \ge 2 - 2^{2-k}$. $\square$

We can now show that when $\mathcal{H}$ is infinite, Proposition 6.3 is tight up to the first term.

Proposition 6.7. *Let $\mathcal{H}$ be an infinite hypothesis class such that $\mathtt{RL}(\mathcal{H}) < \infty$.*

If T is a tree shattered by $\mathcal{H}$ such that $E_T/2 \ge \mathtt{RL}(\mathcal{H}) - \epsilon$, then $\mathsf{depth}(T) \ge \log(1/\epsilon)$.

Proof. Construct a branch $v_0,\ldots,v_\ell$ in T such that for each i, the set of hypotheses $\mathcal{H}(v_i)$ consistent with the path from v_0 to v_i is infinite. This is possible since if $\mathcal{H}(v_i)$ is infinite and v_i is labelled x, then at least one of $\mathcal{H}(v_i)_{x\to 0}, \mathcal{H}(v_i)_{x\to 1}$ is infinite.

Applying Lemma 6.6, we can extend T to another tree T' shattered by $\mathcal{H}$ by hanging from v_ℓ a tree whose expected branch length is arbitrarily close to 2. This shows that

$$\mathtt{RL}(\mathcal{H}) \ge E_{T'}/2 \ge E_T/2 + 2^{-\ell} \ge E_T/2 + 2^{-\mathsf{depth}(T)}. \qquad \square$$

This proposition is tight for the hypothesis class consisting of all $h\colon \mathbb{N} \to \{0,1\}$ such that $|h^{-1}(1)| \le 1$.

We now identify a family of hypothesis classes for which Proposition 6.3 is optimal up to the hidden constant.

Definition 6.8 (Strongly Infinite Hypothesis Class)**.** A hypothesis class $\mathcal{H}$ is *strongly infinite* if it is infinite and for every $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$, the hypothesis class $\mathcal{H}_{x_1 \to y_1, \ldots, x_n \to y_n}$ is either infinite or contains at most one hypothesis.

For example, the hypothesis class consisting of all $h\colon \mathbb{N} \to \{0,1\}$ such that $|h^{-1}(1)| \le k$ is strongly infinite for all $k \ge 1$.

For such classes, we can strengthen Proposition 6.7.

Proposition 6.9. *Let $\mathcal{H}$ be a strongly infinite hypothesis class such that $\mathtt{RL}(\mathcal{H}) < \infty$.*

For every $\epsilon > 0$, any tree T shattered by $\mathcal{H}$ and satisfying $E_T/2 \ge \mathtt{RL}(\mathcal{H}) - \epsilon$ has depth at least $2\mathtt{RL}(\mathcal{H}) + \Omega(\log(1/\epsilon))$.

Proof. Let T be a tree shattered by $\mathcal{H}$ and satisfying $E_T/2 \ge \mathtt{RL}(\mathcal{H}) - \epsilon$.

We can associate each vertex v in T with the example sequence $S(v) = (x_1, y_1), \ldots, (x_r, y_r)$ leading to it. We define $\mathcal{H}(v) = \mathcal{H}_{x_1 \to y_1, \ldots, x_r \to y_r}$.

If v is a leaf of T such that $\mathcal{H}(v)$ is infinite, then we can find an instance x_v such that $S(v), (x_v, 0)$ and $S(v), (x_v, 1)$ are both realizable by $\mathcal{H}$. Let T' be the extension of T obtained by labelling each such leaf v by x_v and adding two leaves. The tree T' is also shattered by $\mathcal{H}$, and so $\mathtt{RL}(\mathcal{H}) \ge E_{T'}/2$. On the other hand, $\mathtt{RL}(\mathcal{H}) \le E_T/2 + \epsilon$.

In order to relate $E_{T'}$ to E_T, let $v_0, \ldots, v_L$ be a random branch in T. Then

$$E_{T'} = E_T + \Pr[\mathcal{H}(v_L) \text{ is infinite}].$$

Since $E_{T'}/2 \le \mathtt{RL}(\mathcal{H}) \le E_T/2 + \epsilon$, this shows that

$$\frac{1}{2} \Pr[\mathcal{H}(v_L) \text{ is infinite}] \le \epsilon. \tag{9}$$

In order to complete the proof, we relate the depth of T to the probability above.

If $i < L$ and $\mathcal{H}(v_i)$ is infinite then $\mathcal{H}(v_{i+1})$ is infinite with probability at least $1/2$. Therefore for every $\ell \in \mathbb{N}$,

$$\Pr[\mathcal{H}(v_L) \text{ is infinite}] \ge \frac{\Pr[L \ge \ell \text{ and } \mathcal{H}(v_\ell) \text{ is infinite}]}{2^{\mathrm{depth}(T) - \ell}}.$$

If $L > \ell$ then $|\mathcal{H}(v_\ell)| \ge 2$, and so $\mathcal{H}(v_\ell)$ is infinite since $\mathcal{H}$ is strongly infinite. Therefore

$$\Pr[\mathcal{H}(v_L) \text{ is infinite}] \ge \frac{\Pr[L > \ell]}{2^{\mathsf{depth}(T) - \ell}}. \tag{10}$$

We can lower bound $\Pr[L > \ell]$ using Markov's inequality:

$$\Pr[L > \ell] = 1 - \Pr[\mathsf{depth}(T) - L \ge \mathsf{depth}(T) - \ell] \ge 1 - \frac{\mathsf{depth}(T) - E_T}{\mathsf{depth}(T) - \ell}.$$

Choosing $\ell = \lfloor 2E_T - \mathsf{depth}(T) \rfloor$, this probability is at least $1/2$, and so Eq. (10) gives

$$\Pr[\mathcal{H}(v_L) \text{ is infinite}] \ge \frac{1}{2^{2\mathsf{depth}(T) - 2E_T + 2}} \ge \frac{1}{2^{2\mathsf{depth}(T) - 4\mathtt{RL}(\mathcal{H}) + 4\epsilon + 2}}.$$

Substituting this in Eq. (9) and rearranging, we conclude that

$$2\mathsf{depth}(T) - 4\mathtt{RL}(\mathcal{H}) - 4\epsilon + 3 \ge \log(1/\epsilon),$$

from which the proposition immediately follows. □

7 Mistake Bounds in the k-Realizable Setting

So far we have considered online learning when the adversary is restricted to choose labels which are consistent with one of the hypotheses in the hypothesis class, a setting known as the *realizable* setting. This is a quite restrictive assumption, and there are many ways to relax it.

In this section we concentrate on the *k-realizable* setting, in which the answers of the adversary are consistent with one of the hypotheses in the class *up to at most k mistakes*. Our goal is to characterize the optimal mistake bounds in this setting, for both deterministic and randomized learners, generalizing Theorems 4.1 and 5.2. Our characterizations are based on *k-shattered trees*, in which each branch is consistent with one of the hypotheses in the class up to at most k mistakes.

If all instances in a sequence of examples are distinct, then the sequence is k-realizable by $\mathcal{H}$ if and only if it is realizable by the *k-expansion* of $\mathcal{H}$, consisting of all hypotheses h' which disagree with some hypothesis $h \in \mathcal{H}$ on at most k instances. However, this need not be the case. For example, the sequence $(x, 0), (x, 1)$ is 1-realizable by the hypothesis class $\mathcal{H}$ consisting of all constant functions.

Nevertheless, the arguments in this section are very similar to their counterparts in the realizable setting.

To complete the picture, we briefly discuss the Perceptron algorithm in this setting in Section 7.7.

7.1 Weighted Hypothesis Classes

While we are interested mainly in the k-realizable setting, we consider a more general setting in which the number of allowed mistakes can depend on the hypothesis. This will be useful in the subsequent proofs.

A weighted hypothesis class $\mathcal{W}$ is a collection of pairs (h, w), where $h\colon \mathcal{X} \to \mathcal{Y}$ is a hypothesis and $w \in \mathbb{N}$ is the allowed number of mistakes (possibly zero). Furthermore, all hypotheses are distinct (that is, $\mathcal{W}$ cannot contain two different pairs $(h, w_1), (h, w_2)$). An input sequence $(x_1, y_1), \ldots, (x_t, y_t)$ is *realizable* by a weighted hypothesis class $\mathcal{W}$ if there exists $(h, w) \in \mathcal{W}$ such that $h(x_i) \neq y_i$ for at most w many examples in the sequence. A tree is *shattered* by $\mathcal{W}$ if each of its branches is realized by $\mathcal{W}$.

Given a hypothesis class $\mathcal{H}$, a learning rule which observes the labeled example (x, y) can restrict itself to $\mathcal{H}_{x \to y} = \{h \in \mathcal{H} : h(x) = y\}$. The corresponding operation for weighted hypothesis classes is

$$\mathcal{W}_{x \to y} = \{(h, w) : (h, w) \in \mathcal{W}, h(x) = y\} \cup \{(h, w - 1) : (h, w) \in \mathcal{W}, h(x) \neq y, w > 0\}.$$

In words, we decrease the allowed number of mistakes for each hypothesis inconsistent with the given example (x, y), removing hypotheses which has zero mistakes left.

For every weighted hypothesis class $\mathcal{W}$, we define its Littlestone dimension and its randomized Littlestone dimension by

$$\mathtt{L}(\mathcal{W}) = \sup_{T \text{ shattered}} m_T \quad \text{and} \quad \mathtt{RL}(\mathcal{W}) = \frac{1}{2} \sup_{T \text{ shattered}} E_T,$$

where the supremum is taken over all trees shattered by $\mathcal{W}$. As in the realizable setting, we define $\mathtt{L}(\emptyset) = \mathtt{RL}(\emptyset) = -1$ for convenience.

Our main results in this section extend Theorems 4.1 and 5.2 to this more general setting.

Theorem 7.1 (Optimal Deterministic Mistake Bound)**.** *Let $\mathcal{W}$ be a weighted hypothesis class. Then,*

$$\mathtt{M}^\star_D(\mathcal{W}) = \mathtt{L}(\mathcal{W}).$$

Theorem 7.2 (Optimal Randomized Mistake Bound)**.** *Let $\mathcal{W}$ be a weighted hypothesis class. Then,*

$$\mathtt{M}^\star(\mathcal{W}) = \mathtt{RL}(\mathcal{W}).$$

We prove these theorems in the following subsections, making use of the following fundamental observation, which follows directly from the definitions:

Observation 7.3. *Let $\mathcal{W}$ be a weighted hypothesis class. The sequence $(x_1, y_1), \dots, (x_t, y_t)$ is realizable by $\mathcal{W}$ iff the sequence $(x_2, y_2), \dots, (x_t, y_t)$ is realizable by $\mathcal{W}_{x_1 \to y_1}$.*

Similarly, let T is a tree whose root is labeled by x, and let T_0, T_1 be the subtrees rooted at the children of the root. Then T is realizable by $\mathcal{W}$ iff T_0 is realizable by $\mathcal{W}_{x \to 0}$ and T_1 is realizable by $\mathcal{W}_{x \to 1}$.

When the weighted hypothesis class is finite, the randomized Littlestone dimension is achieved exactly by some (potentially infinite) tree, as we show in Section 7.6.

The k-realizable setting. Let $\mathcal{H}$ be a hypothesis class, and let $k \in \mathbb{N}$. A sequence of examples $S = \{(x_i, y_i)\}_{i=1}^t$ is *k-realizable* by $\mathcal{H}$ if there exists $h \in \mathcal{H}$ such that $h(x_i) \neq y_i$ for at most k indices i. We denote the corresponding mistake bounds by $\mathtt{M}^\star(\mathcal{H}, k), \mathtt{M}^\star_D(\mathcal{H}, k)$. These are defined just as in the realizable setting, the only difference being that the sequence of examples provided by the adversary need only be k-realizable by $\mathcal{H}$.

We say that a tree is *k-shattered by $\mathcal{H}$* if every branch is k-realizable by $\mathcal{H}$. The corresponding deterministic and randomized k-Littlestone dimension of a class $\mathcal{H}$ are

$$\mathtt{L}_k(\mathcal{H}) = \sup_{T\ k\text{-shattered}} m_T \quad \text{and} \quad \mathtt{RL}_k(\mathcal{H}) = \frac{1}{2} \sup_{T\ k\text{-shattered}} E_T.$$

If we define $\mathcal{W}_{\mathcal{H},k} = \{(h, k) : h \in \mathcal{H}\}$, then a sequence of examples is k-realizable by $\mathcal{H}$ if it is realizable by $\mathcal{W}_{\mathcal{H},k}$. In other words, the k-realizable setting is a special case of weighted hypothesis classes, where all weights are equal to k. Therefore we immediately conclude the following theorems, by applying the preceding theorems to $\mathcal{W}_{\mathcal{H},k}$:

Theorem 7.4 (Optimal Deterministic Mistake Bound)**.** *Let $\mathcal{H}$ be a hypothesis class, and let $k \in \mathbb{N}$. Then,*

$$\mathtt{M}^\star_D(\mathcal{H}, k) = \mathtt{L}_k(\mathcal{H}).$$

Theorem 7.5 (Optimal Randomized Mistake Bound)**.** *Let $\mathcal{H}$ be a hypothesis class, and let $k \in \mathbb{N}$. Then,*

$$\mathtt{M}^\star(\mathcal{H}, k) = \mathtt{RL}_k(\mathcal{H}).$$

Using recent results of [ABED$^+$21], we can bound the optimal mistake bound in terms of the *realizable* Littlestone dimension:

Theorem 7.6. *Let $\mathcal{H}$ be a hypothesis class, and let $k \in \mathbb{N}$. Then,*

$$\mathtt{M}^\star(\mathcal{H}, k) \leq k + O\Big(\sqrt{k \cdot \mathtt{L}(\mathcal{H})} + \mathtt{L}(\mathcal{H})\Big).$$

We prove this result in Section 7.4. Note that since $\mathtt{L}(\mathcal{H})$ and $\mathtt{RL}(\mathcal{H})$ differ by at most a constant factor, the theorem still holds if we replace $\mathtt{L}(\mathcal{H})$ by $\mathtt{RL}(\mathcal{H})$.

Using the experts algorithm of [KvE15], we can construct an algorithm which works in the adaptive setting, that is, without knowledge of k:

Theorem 7.7. *Let $\mathcal{H}$ be a hypothesis class. There is an algorithm* Squint *such that for every input sequence S which is k^*-realizable by $\mathcal{H}$,*

$$\mathtt{M}(\mathsf{Squint}; S) \leq \mathtt{M}^\star(\mathcal{H}, k^*) + O\Big(\sqrt{\mathtt{M}^\star(\mathcal{H}, k^*) \log((k^* + 1) \log \mathtt{M}^\star(\mathcal{H}, k^*))}\Big) .$$

Furthermore, Squint *is adaptive, that is, it has no knowledge of k^*.*

We describe and analyze the algorithm in Section 7.5.

7.2 Proof of Optimal Deterministic Mistake Bound

The case $\mathcal{W} = \emptyset$ holds by definition. Therefore we assume that $\mathcal{W} \neq \emptyset$. The lower bound "$\mathtt{L}(\mathcal{W}) \leq \mathtt{M}^\star_D(\mathcal{W})$" boils down to the following lemma:

Lemma 7.8. *Let $\mathcal{W}$ be a weighted hypothesis class, and let T be a finite tree which is shattered by $\mathcal{W}$. Then, for every deterministic learning rule* Lrn *there exists a realizable sequence S so that $\mathtt{M}(\mathsf{Lrn}; S) \geq m_T$. Furthermore, S corresponds to one of the branches in T.*

Proof. We construct the sequence S by traversing T, starting at the root v_1. At step i, we send Lrn the instance x_i labelling v_i. If the learner predicts $\hat{y}_i$, we set the true label to $y_i = 1 - \hat{y}_i$, and let v_{i+1} be the vertex obtained from v_i by following the leaf labelled y_i. We stop once the process reaches a leaf.

By construction, S corresponds to one of the branches of T, and the number of mistakes is $|S| \geq m_T$. Since T is shattered by $\mathcal{W}$, then S is realizable by $\mathcal{W}$. □

By applying the lemma on every shattered tree and taking the supremum, we conclude the lower bound:

Corollary 7.9 (Lower bound)**.** *For every weighted hypothesis class $\mathcal{W}$ it holds that $\mathtt{M}^\star_D(\mathcal{W}) \geq \mathtt{L}(\mathcal{W})$.*

We now turn to prove the upper bound "$\mathtt{L}(\mathcal{W}) \geq \mathtt{M}^\star_D(\mathcal{W})$". This is achieved via the WeightedSOA learning rule, depicted in Figure 10.

Lemma 7.10 (Upper bound)**.** *Let $\mathcal{W}$ be a non-empty weighted hypothesis class. The* WeightedSOA *learner described in Figure 10 has the mistake bound*

$$\mathtt{M}(\mathsf{WeightedSOA}; S) \leq \mathtt{L}(\mathcal{W})$$

for every input sequence S realizable by $\mathcal{W}$.

Proof. We will show that each time that WeightedSOA makes a mistake, the Littlestone dimension drops by at least 1. That is, if $\hat{y}_i \neq y_i$ then $\mathtt{L}(V^{(i+1)}) < \mathtt{L}(V^{(i)})$. Since the Littlestone dimension is always non-negative, it follows that WeightedSOA makes at most $\mathtt{L}(\mathcal{W})$ mistakes.

Suppose that $\hat{y}_i \neq y_i$ yet $\mathtt{L}(V^{(i+1)}) = \mathtt{L}(V^{(i)})$. The choice of $\hat{y}_i$ shows that $\mathtt{L}(V^{(i)}_{x_i \to 0}) = \mathtt{L}(V^{(i)}_{x_i \to 1}) = \mathtt{L}(V^{(i)})$. This is, however, impossible. Indeed, take trees T_0, T_1 shattering $V^{(i)}_{x_i \to 0}, V^{(i)}_{x_i \to 1}$ with $m_{T_0} = m_{T_1} = \mathtt{L}(V^{(i)})$. Observation 7.3 shows that the tree T whose root is labelled x_i and in which T_0, T_1 are the subtrees of the root's children is shattered by $V^{(i)}$. Since $m_T = \mathtt{L}(V^{(i)}) + 1$, we reach a contradiction. □

```
WeightedSOA

Input: A weighted hypothesis class $\mathcal{W}$.
Initialize: Let $V^{(1)} = \mathcal{W}$.

for $i = 1, 2, \ldots$
    1. Receive $x_i$.
    2. Predict
           $\hat{y}_i = \arg\max_{b \in \mathcal{Y}} \mathtt{L}\Big(V^{(i)}_{x_i \to b}\Big)$.
    3. Receive true label $y_i$.
    4. Update $V^{(i+1)} = V^{(i)}_{x_i \to y_i}$.
```

Figure 10: The weighted version of SOA.

7.3 Proof of Optimal Randomized Mistake Bound

The proof of the optimal mistake bound in the randomized setting, Theorem 7.2, is very similar to the proof of its counterpart in the realizable setting, Theorem 5.2.

The proof of the lower bound "$\mathtt{RL}(\mathcal{W}) \leq \mathtt{M}^\star(\mathcal{W})$" is virtually identical to the proof of Lemma 5.3.

The proof of the upper bound "$\mathtt{RL}(\mathcal{W}) \geq \mathtt{M}^\star(\mathcal{W})$" uses WeightedRandSOA, the weighted counterpart of RandSOA, which appears in Figure 11. The proof of Lemma 5.7 extends, with virtually no changes, to show that $\mathtt{M}(\mathsf{WeightedRandSOA}; S) \leq \mathtt{RL}(\mathcal{W})$ for every input sequence S realizable by $\mathcal{W}$.

7.4 Explicit Upper Bounds in Terms of Littlestone Dimension

Here we prove Theorem 7.6, which bounds $\mathtt{M}^\star(\mathcal{H}, k)$ in terms of k and $\mathtt{L}(\mathcal{H})$ (or $\mathtt{RL}(\mathcal{H})$). In the proof, we use the notation $\mathtt{M}^\star(\mathcal{H}, k, \mathbf{T})$ for the optimal mistake bound when the number of rounds is bounded by $\mathbf{T}$.

[ABED$^+$21] have shown that, for any time horizon $\mathbf{T}$, we always have $\mathtt{M}^\star(\mathcal{H}, k, \mathbf{T}) \leq k + O\Big(\sqrt{\mathbf{T} \cdot \mathtt{L}(\mathcal{H})}\Big)$. By Proposition 6.3, time horizon $\mathbf{T} = 2\mathtt{M}^\star(\mathcal{H}, k) + O(1)$ suffices to guarantee $\mathtt{M}^\star(\mathcal{H}, k) \leq \mathtt{M}^\star(\mathcal{H}, k, \mathbf{T}) + 1$. Plugging this time horizon into the result of [ABED$^+$21] reveals that

$$\mathtt{M}^\star(\mathcal{H}, k) \leq k + O\Big(\sqrt{\mathtt{M}^\star(\mathcal{H}, k) \cdot \mathtt{L}(\mathcal{H})}\Big).$$

Solving this quadratic inequality in $\sqrt{\mathtt{M}^\star(\mathcal{H}, k)}$ yields the inequality claimed in the theorem.

7.5 Adapting to k

This section presents our proof of Theorem 7.7, showing that it is possible to adapt to the value of k without sacrificing too significantly in the expected mistake bound.

The adaptive technique we propose uses an experts algorithm of [KvE15] named Squint, with experts defined by the optimal randomized algorithm for the k-realizable setting, for all values of $k \in \mathbb{N}$ (including $k = 0$).

```
WeightedRandSOA

Input: A weighted hypothesis class $\mathcal{W}$.
Initialize: Let $V^{(1)} = \mathcal{W}$.

for $i = 1, 2, \ldots$
    1. Receive $x_i$.
    2. Predict $p_i \in [0,1]$ such that the value
           $\max\Big\{p_i + \mathtt{RL}\Big(V^{(i)}_{x_i \to 0}\Big), 1 - p_i + \mathtt{RL}\Big(V^{(i)}_{x_i \to 1}\Big)\Big\}$
       is minimized.
    3. Receive true label $y_i$.
    4. Update $V^{(i+1)} = V^{(i)}_{x_i \to y_i}$.
```

Figure 11: The weighted version of RandSOA.

The experts algorithm Squint accepts an input sequence $S = (x_1, y_1), \ldots, (x_n, y_n)$ and a list of learners Lrn_k, each with an associated weight π_k. The weights π_k should form a probability distribution. With an appropriate choice of parameters, Squint has the following guarantee [KvE15, Theorem 3]:

$$\mathtt{M}(\mathsf{Squint}; S) \leq \min_k \left\{ \mathtt{M}(\mathsf{Lrn}_k; S) + O\left(\sqrt{V_k \log \frac{\log V_k}{\pi_k}} + \log \frac{1}{\pi_k} \right) \right\}, \tag{11}$$

where V_k is an uncentered variance term given by

$$V_k = \sum_{i=1}^{n} (|\mathsf{Squint}(x_1, y_1, \ldots, x_{i-1}, y_{i-1}, x_i) - y_i| - |\mathsf{Lrn}_k(x_1, y_1, \ldots, x_{i-1}, y_{i-1}, x_i) - y_i|)^2.$$

Since both absolute values are in the range $[0,1]$, we have

$$\begin{aligned} V_k &\leq \sum_{i=1}^{n} |\mathsf{Squint}(x_1, y_1, \ldots, x_{i-1}, y_{i-1}, x_i) - y_i| + \sum_{i=1}^{n} |\mathsf{Lrn}_k(x_1, y_1, \ldots, x_{i-1}, y_{i-1}, x_i) - y_i| \\ &= \mathtt{M}(\mathsf{Squint}; S) + \mathtt{M}(\mathsf{Lrn}_k; S). \end{aligned}$$

For any given k, if $\mathtt{M}(\mathsf{Squint}; S) > \mathtt{M}(\mathsf{Lrn}_k; S)$, then we have $V_k \leq 2\mathtt{M}(\mathsf{Squint}; S)$, so that (11) implies

$$\mathtt{M}(\mathsf{Squint}; S) \leq \mathtt{M}(\mathsf{Lrn}_k; S) + O\left(\sqrt{\mathtt{M}(\mathsf{Squint}; S) \log \frac{\log \mathtt{M}(\mathsf{Squint}; S)}{\pi_k}} + \log \frac{1}{\pi_k} \right).$$

This inequality trivially holds as well in the case $\mathtt{M}(\mathsf{Squint}; S) \leq \mathtt{M}(\mathsf{Lrn}_k; S)$ due to the first term on the right hand side. Moreover, this inequality further implies

$$\mathtt{M}(\mathsf{Squint}; S) = O\left(\mathtt{M}(\mathsf{Lrn}_k; S) + \log \frac{1}{\pi_k} + 1 \right).$$

To see this, note that were it not the case, we could upper bound each $\mathtt{M}(\mathsf{Lrn}_k; S)$ and $\log \frac{1}{\pi_k}$ on the right hand side by $\mathtt{M}(\mathsf{Squint}; S)/c$ for some large constant c, making the right hand side strictly less than $\mathtt{M}(\mathsf{Squint}; S)$: a contradiction. Plugging in this upper bound on $\mathtt{M}(\mathsf{Squint}; S)$ into the $\log\log \mathtt{M}(\mathsf{Squint}; S)$ term and simplifying with elementary inequalities reveals

$$\mathtt{M}(\mathsf{Squint}; S) \le \mathtt{M}(\mathsf{Lrn}_k; S) + O\left(\sqrt{\mathtt{M}(\mathsf{Squint}; S)\log\frac{\log \mathtt{M}(\mathsf{Lrn}_k; S)}{\pi_k}} + \log\frac{1}{\pi_k}\right).$$

This is a quadratic inequality in $\sqrt{\mathtt{M}(\mathsf{Squint}; S)}$. Solving the quadratic for the range of $\mathtt{M}(\mathsf{Squint}; S)$ where the inequality holds, we have

$$\mathtt{M}(\mathsf{Squint}; S) \le \mathtt{M}(\mathsf{Lrn}_k; S) + O\left(\sqrt{\mathtt{M}(\mathsf{Lrn}_k; S)\log\frac{\log \mathtt{M}(\mathsf{Lrn}_k; S)}{\pi_k}} + \log\frac{\log \mathtt{M}(\mathsf{Lrn}_k; S)}{\pi_k}\right).$$

Since this holds for any k, we conclude that

$$\mathtt{M}(\mathsf{Squint}; S) \le \min_k \left\{\mathtt{M}(\mathsf{Lrn}_k; S) + O\left(\sqrt{\mathtt{M}(\mathsf{Lrn}_k; S)\log\frac{\log \mathtt{M}(\mathsf{Lrn}_k; S)}{\pi_k}} + \log\frac{\log \mathtt{M}(\mathsf{Lrn}_k; S)}{\pi_k}\right)\right\}. \tag{12}$$

We instantiate Squint with algorithm $\mathsf{WeightedRandSOA}$ of Figure 11. Namely, for every k, we let Lrn_k be the instantiation of $\mathsf{WeightedRandSOA}$ with $\mathcal{W}_{\mathcal{H},k}$. We use the weights $\pi_k = \frac{1}{(k+1)(k+2)}$. Since $\pi_k = \frac{1}{k+1} - \frac{1}{k+2}$, they indeed constitute a probability distribution. Since $\mathsf{WeightedRandSOA}$ achieves the optimal mistake bound (see Section 7.3), Eq. (12) shows that if S is k^*-realizable by $\mathcal{H}$ then

$$\mathtt{M}(\mathsf{Squint}; S) \le \mathtt{M}^\star(\mathcal{H}, k^*) + O\left(\sqrt{\mathtt{M}^\star(\mathcal{H}, k^*)\log\big((k^*+1)\log \mathtt{M}^\star(\mathcal{H}, k^*)\big)} + \log\big((k^*+1)\mathtt{M}^\star(\mathcal{H}, k^*)\big)\right).$$

Since $\mathtt{M}^\star(\mathcal{H}, k^*) \ge k^*/2$, the term $\log\big((k^*+1)\mathtt{M}^\star(\mathcal{H}, k^*)\big)$ can be swallowed by the preceding term.

7.6 Finite Classes

The randomized Littlestone dimension is defined as a supremum. The supremum is not always achieved even in the realizable case, as Example 5.23 shows. However, if the hypothesis class is finite, then Proposition 5.22 shows that the randomized Littlestone dimension is achieved by a finite tree.

In this section, we extend the latter result to the setting of weighted hypothesis classes, using infinite trees. The trees that we construct will furthermore "nonredundant", in the following sense.

Definition 7.11 (Nonredundant trees)**.** Let $\mathcal{W}$ be a non-empty weighted hypothesis class, and let T be a non-empty tree shattered by it. The tree T is *weakly nonredundant for* $\mathcal{W}$ if one of the following holds:

1. T is a leaf.

2. The root of T is labelled by an instance x such that either $\mathcal{W}_{x\to 0} \neq \mathcal{W}$ or $\mathcal{W}_{x\to 1} \neq \mathcal{W}$.

3. $\mathcal{W}$ is a singleton (that is, $|\mathcal{W}| = 1$).

A non-empty tree T is *nonredundant for $\mathcal{W}$* if this holds recursively. In detail, if T is a leaf, then it is always nonredundant. Otherwise, let x be the label of the root, leading to the two subtrees T_0, T_1. The tree T is nonredundant for $\mathcal{W}$ if it is weakly nonredundant for $\mathcal{W}$, the tree T_0 is nonredundant for $\mathcal{W}_{x\to 0}$, and tree T_1 is nonredundant for $\mathcal{W}_{x\to 1}$.

If the root of T is labelled by an instance x such that $\mathcal{W}_{x\to 0} = \mathcal{W}$, then this corresponds to an adversarial strategy in which the learner can guarantee that her prediction is correct by predicting 0. Intuitively, there is no reason for the adversary to ask such a question. We prove this formally below.

Proposition 7.12. *Let $\mathcal{W}$ be a finite weighted hypothesis class. There exists a (possibly infinite) tree T_∞ shattered by $\mathcal{W}$ such that*

$$\mathtt{RL}(\mathcal{W}) = \frac{1}{2} E_{T_\infty}.$$

Moreover, T_∞ is monotone and nonredundant for $\mathcal{W}$.

Proof. We start by showing that if we are able to construct a (possibly infinite) tree T_∞ shattered by $\mathcal{W}$ such that $\mathtt{RL}(\mathcal{W}) = E_{T_\infty}/2$, then it is automatically monotone.

If T_∞ is a leaf then it is monotone. Otherwise, suppose that the root is labelled by $x \in \mathcal{X}$, and let the two subtrees of the root be $T_{\infty,0}, T_{\infty,1}$. The subtree $T_{\infty,b}$ must be shattered by $\mathcal{W}_{x\to b}$ and satisfy $\mathtt{RL}(\mathcal{W}_{x\to b}) = E_{T_{\infty,b}}/2$. Clearly $\mathtt{RL}(\mathcal{W}_{x\to b}) \le \mathtt{RL}(\mathcal{W})$, since any tree shattered by $\mathcal{W}_{x\to b}$ is also shattered by $\mathcal{W}$. Therefore $E_{T_{\infty,b}} \le E_{T_\infty}$.

The proof of the rest of the proposition is by induction on the total weight of hypotheses in $\mathcal{W}$. If $\mathtt{RL}(\mathcal{W}) = 0$ then there is nothing to prove. Otherwise, considering all possible roots and using the formula $\mathtt{RL}(\mathcal{W}) = \sup_T E_T/2$, where the supremum is over all trees shattered by $\mathcal{W}$, we see that

$$\mathtt{RL}(\mathcal{W}) = \frac{1}{2} + \sup_{x\in\mathcal{X}} \frac{\mathtt{RL}(\mathcal{W}_{x\to 0}) + \mathtt{RL}(\mathcal{W}_{x\to 1})}{2}.$$

Since $\mathcal{W}$ is finite, there are only finitely many possible pairs $(\mathcal{W}_{x\to 0}, \mathcal{W}_{x\to 1})$. This shows that the supremum is achieved by some instance x, which satisfies

$$\mathtt{RL}(\mathcal{W}) = \frac{1}{2} + \frac{\mathtt{RL}(\mathcal{W}_{x\to 0}) + \mathtt{RL}(\mathcal{W}_{x\to 1})}{2}.$$

If $\mathcal{W}_{x\to 0}, \mathcal{W}_{x\to 1} \neq \mathcal{W}$ then we can apply the induction hypothesis to construct (possibly infinite) nonredundant trees $T_{\infty,0}, T_{\infty,1}$ shattered by $\mathcal{W}_{x\to 0}, \mathcal{W}_{x\to 1}$ (respectively) such that $\mathtt{RL}(\mathcal{W}_{x\to 0}) = E_{T_{\infty,0}}/2$ and $\mathtt{RL}(\mathcal{W}_{x\to 1}) = E_{T_{\infty,1}}/2$. The tree T_∞ comprising a root labelled x leading to the subtrees $T_{\infty,0}, T_{\infty,1}$ then satisfies all the requirements of the proposition.

Suppose next that $\mathcal{W}_{x\to 0} = \mathcal{W}_x$. Calculation shows that

$$\mathtt{RL}(\mathcal{W}) = 1 + \mathtt{RL}(\mathcal{W}_{x\to 1}).$$

We can apply the induction hypothesis to construct a (possibly infinite) nonredundant tree S_∞ shattered by $\mathcal{W}_{x\to 1}$) such that $E_{S_\infty}/2 = \mathtt{RL}(\mathcal{W}_{x\to 1})$. We will show that we can attach to each leaf v of S_∞ a tree T_v satisfying $E_{T_v} = 2$ such that the resulting tree T_∞ is nonredundant and shattered by $\mathcal{W}$. Since $E_{T_\infty}/2 = 1 + E_{S_\infty}/2 = 1 + \mathtt{RL}(\mathcal{W}_{x\to 1}) = \mathtt{RL}(\mathcal{W})$, this will complete the proof.

Let v be a leaf of S_∞, let $(x_1, y_1), \dots, (x_\ell, y_\ell)$ be the path leading to it, and let $\mathcal{W}_v = \mathcal{W}_{x_1\to y_1,\dots,x_\ell\to y_\ell}$. Since $\mathcal{W}_v = \{(h, w+1) : (h, w) \in \mathcal{W}_{x\to 1}\}$ and $\mathcal{W}_{x\to 1}$ is non-empty by construction, we see that $\mathcal{W}_v$ is also non-empty.

We now consider two cases: $\mathcal{W}_v$ is a singleton, and $\mathcal{W}_v$ is not a singleton.

If $\mathcal{W}_v = \{(h,w)\}$ is a singleton, let x be an arbitrary instance, and suppose for concreteness that $h(x) = 0$. We construct a tree T_v which is an infinite left-leaning path (as in Figure 7) in which all vertices are labelled x. The length of a random branch has distribution Geom(1/2), and so $E_{\mathcal{W}_v} = 2$.

If $\mathcal{W}_v$ is not a singleton, let x be an instance such that $h_0(x) = 0$ and $h_1(x) = 1$ for some hypotheses h_0, h_1 in $\mathcal{W}_v$. In this case, T_v is simply the tree consisting of a vertex labelled x leading to two leaves. □

Proposition 7.12 doesn't necessarily hold if we restrict ourselves to finite trees. To see this, consider the weighted hypothesis class $\mathcal{W} = \{(h_0, 1)\}$ over the domain $\mathbb{N}$, in which $h_0(n) = 0$ for all $n \in \mathbb{N}$. All canonical trees shattered by $\mathcal{W}$ are truncations of the infinite path depicted in Figure 7. The infinite path has expected branch length 2, yet its truncation to depth k has expected branch length $2 - 2^{1-k}$.

7.7 The Perceptron

We close this section by considering the classical Perceptron algorithm [Ros58] in the k-realizable setting, showing that its finite mistake-bound guarantee is retained in the k-realizable setting, namely when there exists a linear separator which correctly classifies (with margin) all but k of the examples in the input sequence.

Let us first quickly recall the Perceptron algorithm: its input is a sequence

$$S = (x_1, y_1), \dots, (x_t, y_t),$$

where $x_i \in \mathbb{R}^n$ is the instance and $y_i \in \{\pm 1\}$ is the label. The Perceptron maintains a linear predictor $w_i \in \mathbb{R}^n$, initialized to $w_1 = 0$. Then, at each step i, the Perceptron predicts $\hat{y}_i = \texttt{sign}(\langle w_i, x_i \rangle)$. In case of a mistake, i.e. $\hat{y}_i \neq y_i$, the Perceptron updates its linear predictor by setting $w_{i+1} = w_i + y_i \cdot x_i$. Notice that the Perceptron is mistake-driven, that is, it changes its predictor only when it makes a mistake.

Proposition 7.13 (Perceptron: k-Realizable Mistake Bound)**.** *Assume an input sequence $S = (x_1, y_1), \dots, (x_t, y_t)$ which is k-realizable in the sense that there exists $w \in \mathbb{R}^n$ such that $y_i\langle w, x_i \rangle \geq 1$ for at least $t - k$ indices i. Let*

$$B := \min\{\|w\| : y_i\langle w, x_i\rangle \geq 1 \textit{ for at least } t-k \textit{ indices } i\} \textit{ and } R := \max_i \|x_i\|.$$

Then, the number of mistakes the Perceptron makes on S is at most $B^2R^2 + 2k(BR+1)$.

Proof. The proof is a simple adaptation of the standard analysis. Let M denote the number of mistakes, and let $w^\star = \arg\min\{\|w\| : y_i\langle w, x_i\rangle \geq 1 \text{ for at least } t-k \text{ indices } i\}$, so that $\|w^\star\| = B$. Notice that whenever the Perceptron makes a mistake, it sets w_{i+1} by adding $y_i x_i$ to w_i, where $\langle y_i x_i, w_i\rangle = y_i\langle x_i, w_i\rangle \leq 0$. Thus, the added vector $y_i x_i$ is negatively correlated with w_i, and hence

$$\|w_{i+1}\|^2 \leq \|w_i\|^2 + \|y_i x_i\|^2 \leq \|w_i\|^2 + R^2.$$

Consequently, the final predictor w_t satisfies

$$\|w_t\| \leq \sqrt{M}R. \tag{13}$$

We proceed by lower-bounding $\langle w_t, w^\star\rangle$: consider a step u at which the predictor w_i is being updated (i.e. $\hat{y}_i \neq y_i$). If $y_i\langle w^\star, x_i\rangle \geq 1$ then the standard argument holds:

$$\langle w_{i+1}, w^\star\rangle - \langle w_i, w^\star\rangle = \langle y_i x_i, w^\star\rangle = y_i\langle x_i, w^\star\rangle \geq 1.$$

Otherwise, we use the trivial bound

$$\langle w_{i+1}, w^\star\rangle - \langle w_i, w^\star\rangle = y_i\langle x_i, w^\star\rangle \geq -\|x_i\|\|w^\star\| \geq -BR.$$

Crucially, notice that by k-realizability, the second case (in which $y_i\langle w^\star, x_i\rangle < 1$) happens for at most k steps. Summing up over all the M steps at which there was an update, we get:

$$\langle w_t, w^\star\rangle \geq (M-k)\cdot 1 - k\cdot BR. \tag{14}$$

Combining Equations (13) and (14), we get

$$\sqrt{M}R \geq \|w_t\| \geq \frac{1}{\|w^\star\|}\langle w_t, w^\star\rangle \geq \frac{M-k-kBR}{B}.$$

The latter inequality implies that M satisfies $\sqrt{M}BR \geq M - k(BR+1)$. Squaring, we see that $MB^2R^2 \geq M^2 - 2k(BR+1)M$, and so $M \leq B^2R^2 + 2k(BR+1)$, as required. □

8 Prediction using Expert Advice

In this section, we consider the problem of *prediction using expert advice*, which was raised in [Vov90, LW94]. Specifically, we consider the k-realizable setting, which was suggested in [CBFHW96, CBFH+97] and further studied in [ALW06, MS10, BP19].

The problem concerns a repeated game which has the same flavor as the online learning game of Section 4. The game is between a learner and an adversary. Additionally, there are n experts. Each round i in the game proceeds as follows:

(i) The experts present predictions $\hat{y}_i^{(1)}, \ldots, \hat{y}_i^{(n)} \in \{0,1\}$.

(ii) The learner predicts a value $p_i \in [0,1]$.

(iii) The adversary reveals the true answer $y_i \in \{0,1\}$, and the learner suffers the loss $|y_i - p_i|$.

The adversary must choose the answers so that at least one of the experts makes at most k mistakes. That is, there must exist an expert j such that $y_i \neq \hat{y}_i^{(j)}$ for at most k many indices i. We call such an adversary *k-consistent*.

The goal is to determine the optimal loss of the learner as a function of n and k. We denote the optimal loss of the learner by $\mathtt{M}^\star(n,k)$, and the optimal loss when the learner is constrained to output predictions in $\{0,1\}$ by $\mathtt{M}^\star_D(n,k)$.

The game underlying prediction using expert advice is quite similar to the online learning game. In fact, we can relate the two.

Let $\mathcal{X}_n = \{0,1\}^n$, and consider the hypothesis class $\mathcal{U}_n$ on the domain $\mathcal{X}_n$ consisting of the projection functions $h_i(x_1,\ldots,x_n) = x_i$. We can simulate the game of prediction using expert advice by the online learning game as follows: whenever the experts predict $x_1,\ldots,x_n$, the adversary sends the instance $(x_1,\ldots,x_n)$. The adversary in the original game is k-consistent if and only if the sequence (x_i, y_i) is k-realizable by $\mathcal{U}_n$.

This simulation goes both ways, and so the two games are actually equivalent. The upshot is that we can express $\mathtt{M}^\star(n,k)$ and $\mathtt{M}^\star_D(n,k)$ in terms of quantities we have already considered:

$$\mathtt{M}^\star(n,k) = \mathtt{M}^\star(\mathcal{U}_n,k) = \mathtt{RL}_k(\mathcal{U}_n) \text{ and } \mathtt{M}^\star_D(n,k) = \mathtt{M}^\star(\mathcal{U}_n,k) = \mathtt{L}_k(\mathcal{U}_n).$$

The equivalence above shows that $\mathcal{U}_n$ is the "hardest" hypothesis class of size n, in the sense that it maximizes both $\mathtt{M}^\star(\mathcal{H},k)$ and $\mathtt{M}^\star_D(\mathcal{H},k)$ over all hypothesis classes $\mathcal{H}$ of size n. Indeed, $\mathtt{M}^\star(\mathcal{H},k)$ and $\mathtt{M}^\star_D(\mathcal{H},k)$ are equal to the optimal loss in the game of prediction using expert advice when the answers of the experts must belong to $\{(h_1(x),\ldots,h_n(x)) : x \in \mathcal{X}\}$, where $\mathcal{H} = \{h_1,\ldots,h_n\}$ has domain $\mathcal{X}$.

Bounded horizon. Prediction using expert advice is often considered when the number of rounds is bounded. Let $\mathtt{M}^\star(n,k,\mathbf{T})$ be the optimal loss of the learner when the number of rounds is $\mathbf{T}$.

Clearly $\mathtt{M}^\star(n,k,\mathbf{T}) \le \mathtt{M}^\star(n,k)$. In view of Theorem 6.1, Proposition 6.3 shows that $\mathtt{M}^\star(n,k,\mathbf{T}) \ge \mathtt{M}^\star(n,k) - \epsilon$ already for $\mathbf{T} = 2\mathtt{M}^\star(n,k) + O(\log(1/\epsilon))$. In contrast, since a learner can always guarantee a loss of at most $1/2$ per round by predicting $1/2$, we have $\mathtt{M}^\star(n,k,\mathbf{T}) \le \mathbf{T}/2$, and so $\mathtt{M}^\star(n,k,\mathbf{T}) \ge \mathtt{M}^\star(n,k) - \epsilon$ requires $\mathbf{T} \ge 2\mathtt{M}^\star(n,k) - 2\epsilon$.

(The deterministic case is not interesting, since trivially $\mathtt{M}^\star_D(n,k,\mathbf{T}) = \min\{\mathbf{T}, \mathtt{M}^\star_D(n,k)\}$.)

8.1 Optimal Mistake Bounds

For every $n \ge 1$ and $k \ge 0$, let

$$D(n,k) = \max\left\{ d : d \le \log n + \log \binom{d}{\le k} \right\}.$$

The value of $D(n,k)$ plays a central role in the problem of prediction using expert advice: [CBFHW96] showed that $\mathtt{M}^\star_D(n,k) \le D(n,k)$ using the *Binomial Weights* learning rule, and complemented this with an almost matching lower bound $\mathtt{M}^\star_D(n,k) \ge D(n,k) - O(\log D(n,k))$. The lower bound is proved by constructing a k-covering code of size n that simulates the experts. When k is fixed, it can be further improved to $\mathtt{M}^\star_D(n,k) \ge D(n,k) - c(k)$, where $c(k)$ is a constant depending on k, by constructing a better covering code[13] [CHLL97, Theorem 12.4.3]. We sketch good approximations to $D(n,k)$ in Section 8.4.

The paper [CBFHW96] leaves open the problem of determining $\mathtt{M}^\star(n,k)$. An extension of Proposition 5.17 shows that $\mathtt{M}^\star(n,k) \le \mathtt{M}^\star_D(n,k) \le 2\mathtt{M}^\star(n,k)$. [ALW06] showed that for large n (as a function of k), the second bound is almost tight: $\mathtt{M}^\star(n,k) \le \mathtt{M}^\star_D(n,k)/2 + O(1)$. Using our approximations for $D(n,k)$, one can see that [BP19] showed that for $k = o(\log n)$, $\mathtt{M}^\star(n,k) \le D(n,k)/2 + o(D(n,k))$. This result applies even in the multiclass setting where the experts' predictions are chosen from some finite set $\{1,\dots,d\}$.

In this section, we remove the assumption that n is large enough, proving the following theorem:

Theorem 8.1. *Let $n \ge 2$ and $k \ge 0$. Then*

$$\mathtt{M}^\star(n,k) \le D(n,k)/2 + O(\sqrt{D(n,k)}).$$

The error term is tight for $n = 2$:

Theorem 8.2. *Let $k \ge 0$. Then*

$$\mathtt{M}^\star(2,k) = D(2,k)/2 + \Omega(\sqrt{D(2,k)}).$$

We also quantitatively improve the results of [ALW06] for large n:

Theorem 8.3. *Let $n \ge 2$, and suppose that $k \le c \log n$ for some $c < 1/2$. Then there exists a constant C, depending only on c, such that*

$$\mathtt{M}^\star(n,k) \le D(n,k)/2 + C.$$

[13] In this case, the construction provides an explicit code, namely a direct sum of k many *Hamming codes* [Ham50].

This result shows that it suffices to have n exponential in k in order to get a bound of the form $D(n,k)/2 + O(1)$.

All of our bounds are attained using the randomized k-Littlestone dimension of $\mathcal{U}_n$. We prove the upper bounds in Section 8.2, and the lower bound in Section 8.3. We close the section by describing an efficient randomized algorithm in the perfect expert setting ($k = 0$) in Section 8.5, and by proving that the optimal learning rule in that case is necessarily improper in Section 8.6.

All results we stated so far concern $n \geq 2$. The case $n = 1$ is different, and much simpler:

Theorem 8.4. *Let $k \geq 0$. Then*

$$\mathtt{M}^\star(1,k) = \mathtt{M}^\star_D(1,k) = D(1,k) = k.$$

Proof. According to the definition, $D(1,k)$ is the maximum d such that $2^d \leq \binom{d}{\leq k}$. Since $\binom{d}{\leq d} = 2^d$ whereas $\binom{d+1}{\leq d} < 2^{d+1}$, we see that $D(1,k) = k$.

The complete tree of depth k, labelled arbitrarily, is k-shattered by $\mathcal{U}_1$. In contrast, a tree of depth $k+1$ cannot be k-shattered by $\mathcal{U}_1$, since there exists a branch on which the unique hypothesis makes $k+1$ mistakes. Therefore $\mathtt{M}^\star_D(1,k) = k$.

For the randomized case, according to Proposition 7.12 there is an infinite tree T_k such that $\mathtt{M}^\star(1,k) = \mathtt{RL}_k(\mathcal{U}_1) = E_{T_k}/2$. Denote the unique hypothesis in $\mathcal{U}_1$ by h. By possibly switching the order of children, we can assume that all vertices in T_k are labelled by an instance x such that $h(x) = 0$. We can then identify vertices of T_k with binary strings.

Since T_k is optimal, it contains all strings which contain at most k many 1s. A string is a leaf it it contains exactly k many 1s and it ends with 1. The length of a random branch has the distribution of a sum of k many Geom(1/2) random variables, and so $\mathtt{M}^\star(1,k) = E_{T_k}/2 = k$. □

In contrast, [LW94] shows that $\mathtt{M}^\star_D(n,k) \geq 2k + \lfloor \log n \rfloor$ for $n \geq 2$, highlighting the difference between $n = 1$ and $n > 1$. This immediately implies the following corollary, which will be useful in the sequel:

Corollary 8.5. *Let $n \geq 2$ and $k \geq 0$. Then $D(n,k) \geq 2k + 1$.*

Proof. Clearly $D(n,k) \geq D(2,k)$. Theorem 8.7 shows that $D(2,k) \geq M^\star_D(2,k)$, which is at least $2k+1$ by the result of [LW94]. □

8.2 Proofs of the Upper Bounds on $\mathtt{M}^\star(n,k)$

We start by proving a probabilistic version of the *sphere packing bound* for covering codes [CHLL97].

Lemma 8.6. *Let $\mathcal{H}$ be a finite hypothesis class of size $n \geq 1$. Let $t \geq k \geq 0$, and let T be a tree whose minimum depth is at least t.*

Let $S = (x_1, y_1), \dots, (x_t, y_t)$ be the random prefix *of length t, consisting of the first t steps in a random branch of T. The probability that S is k-realizable by $\mathcal{H}$ is at most*

$$n\binom{t}{\leq k}/2^t.$$

Proof. For each hypothesis $h \in \mathcal{H}$ and set of indices $I \subseteq [t]$, the probability that $y_i \neq h(x_i)$ for all indices in I and $y_i = h(x_i)$ for all indices outside of I is 2^{-t}.

The sequence S is k-realizable by $\mathcal{H}$ if the event above happens for some $h \in \mathcal{H}$ and some I of size at most k. Applying the union bound, we get that the probability is at most $n\binom{t}{\leq k}/2^t$. □

As a warm-up, we use this lemma together with the k-Littlestone dimension to reprove the upper bound $\mathtt{M}^\star_D(n,k) \leq D(n,k)$, first proved in [CBFHW96].

Theorem 8.7. *Let $n \geq 1$ and $k \geq 0$. Then $\mathtt{M}^\star_D(n,k) \leq D(n,k)$.*

Proof. Since $\mathtt{M}^\star_D(n,k) = \mathtt{L}_k(\mathcal{U}_n)$, it suffices to bound $\mathtt{L}_k(\mathcal{U}_n)$.

Let T be a tree satisfying $m_T = \mathtt{L}_k(\mathcal{U}_n)$ which is k-shattered by $\mathcal{U}_n$. A random prefix of length $\mathtt{L}_k(\mathcal{U}_n)$ is k-realizable by $\mathcal{U}_n$, and so $2^{\mathtt{L}_k(\mathcal{U}_n)} \leq n\binom{\mathtt{L}_k(\mathcal{U}_n)}{\leq k}$ by Lemma 8.6. Taking the logarithm, we deduce that $\mathtt{L}_k(\mathcal{U}_n) \leq D(n,k)$ by the definition of $D(n,k)$. □

We now prove Theorem 8.1 and Theorem 8.3. The main tools are concentration of the random branch length in quasi-balanced trees (Lemma 6.2), and the following lemma.

Lemma 8.8. *Let $\mathcal{H}$ be a finite hypothesis class of size $n \geq 1$. Let $D = D(n,k)$, and let T be a tree of minimum depth at least $(1+\epsilon)D$, where $0 < \epsilon < 1/3$. The probability that a random prefix of length $(1+\epsilon)D$ is k-realizable by $\mathcal{H}$ is at most*

$$2^{1-\epsilon^2 D/9}.$$

Furthermore, if $k \leq c\log n$ for some constant $c < 1/2$ then the probability is at most

$$2^{1-c'\epsilon D},$$

where $c' > 0$ is a constant depending only on c.

The proof of this lemma will require some elementary estimates on binomial coefficients, summarized in the following technical lemma.

Lemma 8.9. *Let $D \geq k \geq 1$ and $\epsilon > 0$. Then*

$$\binom{(1+\epsilon)D}{\leq k} \leq 2^{\epsilon D \cdot \log(D/(D-k))} \cdot \binom{D}{\leq k}.$$

If furthermore $k \leq D/2$ and $\epsilon \leq 1/3$ then

$$\binom{(1+\epsilon)D}{\leq k} \leq 2^{\epsilon D - \epsilon^2 k/3} \cdot \binom{D}{\leq k}.$$

We prove this lemma in Subsection 8.2.1.

Proof of Lemma 8.8. We start by observing that

$$n\binom{D}{\leq k}/2^D \leq 2, \tag{15}$$

Indeed, the maximality of D shows that

$$1 > n\binom{D+1}{\leq k}/2^{D+1} \geq \frac{1}{2}n\binom{D}{\leq k}/2^D,$$

from which Eq. (15) immediately follows.

Denote by p the probability we wish to bound. Lemma 8.6 shows that

$$p \leq n\binom{(1+\epsilon)D}{\leq k}/2^{(1+\epsilon)D} = \frac{\binom{(1+\epsilon)D}{\leq k}}{\binom{D}{\leq k}} \cdot 2^{-\epsilon D} \cdot n\binom{D}{\leq k}/2^D \leq 2^{1-\epsilon D} \cdot \frac{\binom{(1+\epsilon)D}{\leq k}}{\binom{D}{\leq k}},$$

using Eq. (15). It remains to estimate the ratio using Lemma 8.9.

We start by proving the "furthermore" part. The definition of D implies that $D \geq \log n$, and so $k \leq cD$. Applying Lemma 8.9, we deduce that

$$p \leq 2^{1-(1-\log(D/(D-k)))\epsilon D}.$$

Since

$$c' = 1 - \log \frac{D}{D-k} = 1 - \log \frac{1}{1-k/D} \geq 1 - \log \frac{1}{1-c} > 0,$$

this completes the proof of the "furthermore" part.

In order to prove the main part of the lemma, we distinguish between two cases. If $k \leq D/3$ then the "furthermore" bound shows that

$$p \leq 2^{1-c'\epsilon D},$$

where $c' = \log(4/3)$. Since $\epsilon \leq 1/3$, we have $c'\epsilon \geq \epsilon^2/9$, completing the proof in this case.

Otherwise, $k \geq D/3$. In this case, noting that $k \leq D/2$ by Corollary 8.5, we apply the "furthermore" part of Lemma 8.9 to obtain

$$p \leq 2^{1-\epsilon^2 k/3} \leq 2^{1-\epsilon^2 D/9}. \qquad \square$$

We can now prove the upper bounds on $\mathtt{M}^\star(n,k)$. The idea is simple. Let T be a tree which is k-shattered by $\mathcal{U}_n$. Using Proposition 7.12, we can assume that T is quasi-balanced, and so the length of a random branch is concentrated around E_T. This implies that T realizes almost all sequences of size $(1-\epsilon)E_T$. These sequences are k-realized by $\mathcal{U}_n$, and we obtain an upper bound on E_T via Lemma 8.8.

Proof of Theorem 8.1. Since $\mathtt{M}(n,k) = \mathtt{RL}_k(\mathcal{U}_n)$, we bound the latter. Proposition 7.12 shows that there is an infinite tree T which is k-shattered by $\mathcal{U}_n$ and satisfies $E_T/2 = \mathtt{RL}_k(\mathcal{U}_n)$. Furthermore, T is monotone, and so Proposition 6.2 applies to it (while the proposition is formulated for finite quasi-balanced trees, the proof actually directly uses monotonicity, and is valid for infinite trees).

In order to bound E_T, we will show that for small enough $\epsilon > 0$, the assumption $(1+\epsilon)D \leq (1-\epsilon)E_T$ leads to a contradiction.

Extend T arbitrarily to a tree T' of minimum depth $(1+\epsilon)D$, and let S be a random prefix of T' of length $(1+\epsilon)D$. If S lies completely within T then it is k-realizable by $\mathcal{U}_n$, and so

$$\Pr[S \text{ lies within } T] \leq \Pr[S \text{ is } k\text{-realizable by } \mathcal{U}_n].$$

The probability that S lies within T is precisely the probability that a random branch of T has length at least $(1+\epsilon)D$. Since we assume that $(1+\epsilon)D \leq (1-\epsilon)E_T$, this probability is at least $1 - e^{-\epsilon^2 E_T/4}$ by Proposition 6.2, and so at least $1 - e^{-\epsilon^2 D/4}$.

In contrast, the probability that S is k-realizable by $\mathcal{U}_n$ is at most $2^{1-\epsilon^2 D/9}$ by Lemma 8.8. Therefore

$$1 \leq e^{-\epsilon^2 D/4} + 2^{1-\epsilon^2 D/9}.$$

Let $\epsilon = C/\sqrt{D}$. As $C \to \infty$, the right-hand side tends to 0, and in particular, we obtain a contradiction for some constant $C > 0$.

It follows that $(1+\epsilon)D > (1-\epsilon)E_T$ for $\epsilon = C/\sqrt{D}$, and so

$$E_T < \frac{1+\epsilon}{1-\epsilon} D = (1 + O(1/\sqrt{D}))D = D + O(\sqrt{D}). \qquad \square$$

The proof of Theorem 8.3 is similar, and uses the "furthermore" clause of Lemma 8.8.

Proof of Theorem 8.3. We closely follow the proof of Theorem 8.1, and we only indicate the part which is different.

We start by assuming that $(1+\epsilon)D \leq (1-\epsilon)E_T$ for some $\epsilon > 0$. The assumption on k implies that we can use the "furthermore" clause of Lemma 8.8, and so for some constant $c' > 0$ depending on c,

$$1 \leq e^{-\epsilon^2 D/4} + 2e^{-c'\epsilon D}.$$

Let $\epsilon = C/D$, where C is chosen so that $2e^{-c'\epsilon D} \leq 1/2$. Therefore

$$1/2 \leq e^{-C^2/4D},$$

which fails for $D \geq C'$, where C' depends only on c.

We conclude that if $D \geq C'$ then

$$E_T < \frac{1+\epsilon}{1-\epsilon} D = (1 + O(1/D))D = D + O(1).$$

If $D < C'$, then this follows from the bound $E_T \leq 2\mathtt{M}^*(n,k) \leq 2\mathtt{M}^*_D(n,k) \leq 2D$, where we used an appropriate extension of Proposition 5.17 together with Theorem 8.7. □

8.2.1 Proof of Technical Estimate

In this section we complete the proofs of Theorem 8.1 and Theorem 8.3 by proving Lemma 8.9.

We start with estimates on the ratio of individual binomial coefficients.

Lemma 8.10. *Let $D \geq \ell \geq 1$ and $\epsilon > 0$. Then*

$$\binom{(1+\epsilon)D}{\ell} \leq 2^{\epsilon D \cdot \log(D/(D-\ell))} \cdot \binom{D}{\ell}.$$

If furthermore $\ell \leq D/2$ and $\epsilon \leq 1/3$ then

$$\binom{(1+\epsilon)D}{\ell} \leq 2^{\epsilon D \cdot \log(D/(D-\ell)) - \epsilon^2 \ell/3} \cdot \binom{D}{\ell}.$$

Proof. We can calculate the ratio between the binomials explicitly:

$$R_\ell := \binom{(1+\epsilon)D}{\ell} \Big/ \binom{D}{\ell} = \prod_{r=0}^{\ell-1} \frac{(1+\epsilon)D - r}{D-r} = \prod_{r=0}^{\ell-1} \left(1 + \frac{\epsilon D}{D-r}\right).$$

Applying the well-known estimate $\ln(1+x) \leq x$, we obtain

$$\ln R_\ell \leq \sum_{r=0}^{\ell-1} \frac{\epsilon D}{D-r} \leq \epsilon D \cdot \int_{D-\ell}^{D} \frac{dx}{x} = \epsilon D \cdot \ln \frac{D}{D-\ell},$$

and so

$$R_\ell \leq 2^{\epsilon D \cdot \log(D/(D-\ell))}.$$

Now suppose that $\ell \leq D/2$ and $\epsilon \leq 1/3$. For $r \in \{0, \ldots, \ell-1\}$ we have

$$\frac{\epsilon D}{D-r} \leq \frac{\epsilon D}{D-\ell} = \frac{\epsilon}{1-\ell/D} \leq 2\epsilon \leq 2/3.$$

Since $1 + x \leq e^{x - x^2/3}$ for $x \leq 0.787$, we can improve the estimate on R_ℓ:

$$\ln R_\ell \leq \epsilon D \cdot \ln \frac{D}{D-\ell} - \frac{1}{3} \sum_{r=0}^{\ell-1} \frac{\epsilon^2 D^2}{(D-r)^2} \leq \epsilon D \cdot \ln \frac{D}{D-\ell} - \frac{1}{3}\epsilon^2 \ell. \qquad \square$$

We can now prove Lemma 8.9.

Proof of Lemma 8.9. The ratio between $\binom{(1+\epsilon)D}{\leq k}$ and $\binom{D}{\leq k}$ is clearly at most $\max(R_0, \ldots, R_k)$, where R_ℓ is the ratio between the binomials in Lemma 8.10.

If we only assume that $D \geq \ell \geq 1$ and $\epsilon > 0$, then Lemma 8.10 states that

$$\log R_\ell \leq \epsilon D \cdot \log \frac{D}{D-\ell},$$

which is clearly monotone increasing in ℓ. Therefore

$$\log \max(R_0, \ldots, R_k) \leq \epsilon D \cdot \log \frac{D}{D-k}.$$

If we furthermore assume that $k \leq D/2$ and $\epsilon \leq 1/3$, then Lemma 8.10 states that

$$\log R_\ell \leq \epsilon D \cdot \log \frac{D}{D-\ell} - \frac{1}{3}\epsilon^2 \ell.$$

The derivative of the upper bound with respect to ℓ is

$$\frac{\epsilon D}{D-\ell} - \frac{1}{3}\epsilon^2 \geq \epsilon - \frac{1}{3}\epsilon^2 > 0,$$

since $\epsilon \leq 1/3$. Therefore the upper bound is maximized at $\ell = k$, and we conclude that

$$\log \max(R_0, \ldots, R_k) \leq \epsilon D \cdot \log \frac{D}{D-k} - \frac{1}{3}\epsilon^2 k.$$

Since $k \leq D/2$, we can further estimate

$$\log \frac{D}{D-k} = \log \frac{1}{1-k/D} \leq \log 2 = 1. \qquad \square$$

8.3 Proof of the Lower Bound on $\mathtt{M}^\star(2,k)$

In order to prove Theorem 8.2, we will compute $\mathtt{M}^\star(2,k)$ *exactly.*

Theorem 8.11. *For all $k \geq 0$,*

$$\mathtt{M}^\star(2,k) = k + \frac{(k+1/2)\binom{2k}{k}}{4^k} \text{ and } \mathtt{M}^\star_D(2,k) = D(2,k) = 2k+1.$$

Well-known estimates of central binomial coefficients state that $\binom{2k}{k} = \Theta(4^k/\sqrt{k})$, see for example [Ele14], and so

$$\mathtt{M}^\star(2,k) \geq k + \Omega(\sqrt{k}) \geq D(2,k)/2 + \Omega(\sqrt{D(2,k)}).$$

Proof of Theorem 8.11. According to Proposition 7.12, there is a nonredundant infinite tree T which is k-shattered by $\mathcal{U}_2$ and satisfies $\mathtt{M}^\star(2,k) = \mathtt{RL}_k(\mathcal{U}_2) = E_T/2$. We will show that without loss of generality, all vertices in T are labelled $(0,1)$. This will allow us to determine T exactly, and so to compute $\mathtt{M}^\star(2,k)$.

If there is a vertex labelled $(1,0)$, we can switch its label to $(0,1)$ and switch its two children. The resulting tree is also k-shattered by $\mathcal{U}_2$ and has the same expected branch length.

If a vertex is labelled $(0,0)$, then by nonredundancy, only one hypothesis is "still in play" (that is, all branches passing through the vertex are realized by the same hypothesis), say the first one. Therefore if we change its label to $(0, 1)$ then the resulting tree is also k-shattered by $\mathcal{U}_2$.

Concluding, we can assume without loss of generality that all vertices in T are labelled $(0, 1)$. Such a tree is k-shattered by $\mathcal{U}_2$ if every prefix (path starting at the root) contains at most k many 0-edges or at most k many 1-edges. Identifying vertices in the tree by the strings formed from the labels of the edges leading to them from the root, the labels of all vertices must contain at most k many 0s or at most k many 1s. We call such strings *legal*.

Since the tree T is optimal, its leaves correspond to legal strings s such that either $s0$ or $s1$ is illegal. If s terminates with 0 then it is a leaf if it either contains at least $k+1$ many 1s and exactly k many 0s (in which case $s0$ is illegal), or if it contains exactly k many 1s and exactly $k+1$ many 0s (in which case $s1$ is illegal). This defines T completely, and we can calculate

$$E_T = 2\sum_{t=k+1}^{\infty}(t+k)\frac{\binom{t+k-1}{k-1}}{2^{t+k}} + 2\cdot(2k+1)\frac{\binom{2k}{k}}{2^{2k+1}} = 2k\sum_{t=k+1}^{\infty}\frac{\binom{t+k}{k}}{2^{t+k}} + \frac{(2k+1)\binom{2k}{k}}{4^k}.$$

The infinite series is the probability that if we toss an unbiased coin, then eventually both sides show up at least $k+1$ many times (if the last toss was heads then t is the number of tails, and vice versa). Therefore

$$E_T = 2k + \frac{(2k+1)\binom{2k}{k}}{4^k}.$$

The formula for $\mathtt{M}^\star(2, k)$ immediately follows.

All leaves in T have depth at least $2k+1$, and so

$$\mathtt{M}^\star_D(2, k) = \mathtt{L}_k(\mathcal{U}_2) \geq m_T = 2k+1.$$

Conversely, suppose that T' is a tree in which each leaf has depth at least $2k+2$. Truncate it so that each leaf has depth exactly $2k+2$. Construct a branch by first following $k+1$ times the edges which disagree with the value of one hypothesis in $\mathcal{U}_2$, and then following $k+1$ the edges which disagree with the value of the other hypothesis in $\mathcal{U}_2$. The resulting branch shows that T' is not k-realizable by $\mathcal{U}_2$, and so $m_{T'} \leq 2k+1$ for any tree T' which is k-realizable by $\mathcal{U}_2$, implying that $\mathtt{M}^\star_D(2, k) = \mathtt{L}_k(\mathcal{U}_2) = 2k+1$.

Finally, let us determine $D(2, k)$. If $d \geq 2k+1$ then

$$\log 2 + \log\binom{d}{\leq k} \leq \log 2 + \log 2^{d-1} = d,$$

with equality if and only if $d = 2k+1$. Therefore $D(2, k) = 2k+1$. □

8.4 Approximations of $D(n, k)$

The quantity $D(n, k)$ appears in the bounds on both $\mathtt{M}^\star(n, k)$ and $\mathtt{M}^\star_D(n, k)$. In this brief section, we sketch how to approximate $D(n, k)$, leaving the grueling details for a future occasion.

Recall that $D(n, k)$ is the maximal d such that $d \leq \log n + \log\binom{d}{\leq k}$. It is well-known that $\binom{d}{\leq k} \leq 2^{dh(k/d)}$, where $h(p) = -p\log p - (1-p)\log(1-p)$ is the binary entropy function. Moreover, this approximation is tight up to a factor of $O(\sqrt{d})$ [Wik22]. Using this approximation, we see that $D(n, k)$ is approximately the solution of the equation

$$d = \log n + dh(k/d).$$

Regime	Approximation
$k = o(\log n)$	$D(n,k) \approx \log n + k \log\left(\frac{\log n}{k}\right)$
$k = \frac{\log n}{c}$ for constant c	$D(n,k) \approx k/f^{-1}(c)$
$k = \omega(\log n)$	$D(n,k) \approx 2k + 2\sqrt{k \ln n}$

Table 2: Approximations of $D(n,k)$ in various regimes

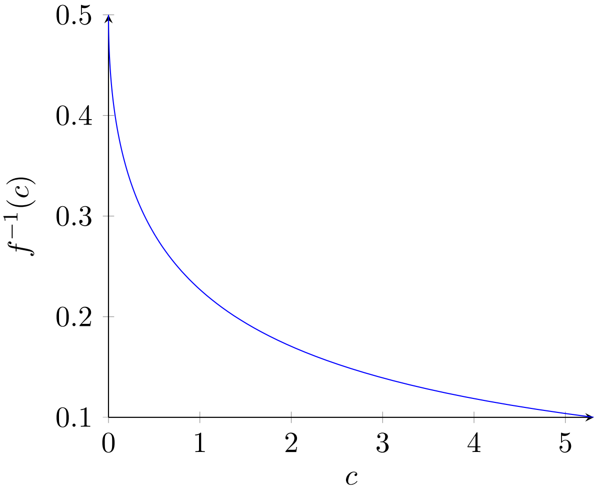

Figure 12: Plot of $f^{-1}(c)$, where $f(p) = (1 - h(p))/p$

Rearranging,

$$\frac{1 - h(k/d)}{k/d} = \frac{\log n}{k}.$$

Therefore, if we define

$$f(p) = \frac{1 - h(p)}{p}$$

then, roughly speaking,

$$D(n,k) \approx \frac{k}{f^{-1}\left(\frac{\log n}{k}\right)},$$

where we take the branch of the inverse which lies in $(0, 1/2]$. Using this, we can derive the approximations appearing in Table 2. The function $f^{-1}(c)$ is plotted in Figure 12.

In the literature on prediction using expert advice, some papers obtain bounds in terms of $D(n,k)$ or variations of it [CBFHW96,MS10], while others give explicit bounds [CBFH$^+$97,BP19].

8.5 Improved Results for the “Perfect Expert” Case

In this section, we study the optimal loss of the learner in prediction using expert advice when the experts are not allowed to make any mistakes, that is, when $k = 0$. This setting is known as the *perfect expert* case. Equivalently, we determine $\mathtt{M}^\star(\mathcal{U}_n)$ exactly.

In the deterministic case, we have $\mathtt{M}^\star_D(\mathcal{U}_n) = \mathtt{L}(\mathcal{U}_n) = \lfloor \log n \rfloor$ [LW94]. In the randomized case, [CBFH$^+$97, BP19] proved that $\mathtt{M}^\star(\mathcal{U}_n) = \log n/2 \pm O(1)$. We determine the exact value of $\mathtt{M}^\star(\mathcal{U}_n)$. We also give an optimal learner achieving exactly the mistake bound $\mathtt{M}^\star(\mathcal{U}_n)$ which is efficient.

We derive our results by a relatively simple analysis of $\mathtt{RL}(\mathcal{U}_n)$. The key is showing that the unique optimal tree shattered by $\mathcal{U}_n$ has exactly n leaves, and its structure is as balanced as possible. This reduces the problem to calculating the expected branch length of such a tree, and therefore the rest is just relatively straightforward calculations (a calculation in the same spirit appears in [DFGM17]).

We say that a tree is *balanced* if its leaves lie in at most two different depths. The following lemma shows that the tree realizing $\mathtt{RL}(\mathcal{U}_n)$ is balanced.

Lemma 8.12. *Let $n = (1+\alpha)2^a \in \mathbb{N}$, where $a \in \mathbb{N}$ and $0 \le \alpha < 1$. There exists a balanced tree T_n having n leaves, which is shattered by $\mathcal{U}_n$, and such that $\mathtt{RL}(\mathcal{U}_n) = E_{T_n}/2$.*

Proof. Since $\mathcal{U}_n$ is finite, Proposition 5.22 shows that there exists some tree T_n such that $\mathtt{RL}(\mathcal{U}_n) = E_{T_n}/2$.

Since different branches disagree on the value of at least one instance, all leaves of T_n must be realized by different hypotheses in $\mathcal{U}_n$. Therefore T_n contains at most n leaves.

Conversely, every unlabelled tree with at most n leaves can be labelled so that it is shattered by $\mathcal{U}_n$. To see this, annotate the leaves of the tree with distinct hypotheses of $\mathcal{U}_n$. Given a vertex v, we label the vertex with an instance which assigns 0 to the hypotheses in its left subtree and 1 to the hypotheses in its right subtree.

Given an unlabelled tree with fewer than n leaves, we can increase its number of leaves by splitting a leaf. Since this operation increases the expected branch length, we see that T_n necessarily contains exactly n leaves.

It remains to show that T_n is balanced. Suppose that T_n is unbalanced, say having a leaf u at depth d and an internal vertex v at depth $D > d$ whose two children are leaves. Consider the unlabelled tree T'_n formed by making v a leaf and adding two children to u. The tree T'_n has the same number of leaves as T_n, and

$$E_{T'_n} = E_{T_n} - \frac{1}{2^D} + \frac{1}{2^d} > E_{T_n}.$$

Indeed, a random branch that reaches v (which happens with probability $1/2^D$) is now one edge shorter, while a random branch that reaches u (which happens with probability $1/2^d$) is one edge longer. Since T_n maximizes E_{T_n} over all unlabelled trees with n leaves, it must be balanced. □

This allows us to determine the exact mistake bound.

Lemma 8.13 (Exact mistake bound)**.** *Let $n = (1+\alpha)2^a \in \mathbb{N}$, where $a \in \mathbb{N}$ and $0 \le \alpha < 1$. Then*

$$\mathtt{M}^\star(\mathcal{U}_n) = \frac{a+\alpha}{2}.$$

Proof. Lemma 8.12 shows that $\mathtt{M}^\star(\mathcal{U}_n) = \mathtt{RL}(\mathcal{U}_n) = E_{T_n}/2$, where T_n is a balanced tree with n leaves.

We claim that T_n has depth a. Indeed, a balanced tree of minimum depth at most $a-1$ has fewer than $2^a \le n$ leaves, and a balanced tree of minimum depth at least $a+1$ has at least $2^{a+1} > n$ leaves. If there are $2^a - B$ leaves at depth a then there are $2B$ leaves at depth $a+1$, and so $2^a + B$ leaves in total. Solving for B, we find that $B = \alpha 2^a$. Therefore

$$E_{T_n} = (1-\alpha)2^a \cdot \frac{a}{2^a} + 2\alpha 2^a \cdot \frac{a+1}{2^{a+1}} = a + \alpha. \qquad \square$$

This lemma can be used to implement RandSOA efficiently in the “perfect expert” setting.

8.6 Proper Learners are Sub-Optimal

It is natural to ask for a learning rule to be *proper*.

Definition 8.14 (Online proper learners [HLM21])**.** Let $\mathcal{H}$ be a concept class. An online learning rule Lrn is *proper* for $\mathcal{H}$ if for every realizable input sequence S, the function $h_S\colon \mathcal{X} \to [0,1]$ given by

$$h_S(x) = \mathsf{Lrn}(S, x)$$

is a convex combination of hypotheses from $\mathcal{H}$, that is, there are coefficients α_h such that

$$h_S(x) = \sum_{h \in \mathcal{H}} \alpha_h h(x).$$

When the learner is deterministic, the function h_S is $\{0,1\}$-valued, and so the learner is proper if $h_S \in \mathcal{H}$ for every realizable input sequence S.

We can adapt this definition to the setting of prediction using expert advice (with $k = 0$) by requiring that at all times, the learner picks a convex combination of the experts before seeing their current advice. In other words, each round of the game is played as follows:

(i) The algorithm chooses a convex combination of the experts.

(ii) The adversary chooses both the advice of the experts and the correct label.

This can also be expressed in the language of game theory: in each round, the first player (the learner) picks a mixed strategy (a convex combination of experts), and then the second player (the adversary) picks a pure strategy (the true label). The payoff is the probability that the learner's random strategy agrees with the adversary's pure strategy.

For a hypothesis class $\mathcal{H}$ we define the optimal randomized mistake bound for proper learners $\mathtt{M}^\star_p(\mathcal{H})$ by restricting the learners to be proper:

$$\mathtt{M}^\star_p(\mathcal{H}) = \inf_{\mathsf{Lrn}_p} \sup_{S} \mathtt{M}(\mathsf{Lrn}_p; S),$$

where the infimum is taken over all proper learning rules, and the supremum is taken over all realizable sequences.

We can similarly define the analogous notion for prediction using expert advice, namely $\mathtt{M}^\star_p(n) = \mathtt{M}^\star_p(\mathcal{U}_n)$.

We can solve the problem of prediction using expert advice optimally with the learning rule $\mathsf{RandSOA}$. This learning rule is improper, a property it shares with Littlestone's SOA algorithm that it is based on. In this section, we show that any proper learning rule makes more mistakes than $\mathsf{RandSOA}$ when used to solve this problem.

Theorem 8.15 (Mistake bound of a proper learner)**.** *For every $n \geq 1$, the optimal mistake bound for proper randomized learners solving prediction using expert advice is*

$$\mathtt{M}^\star_p(n) = H_n - 1 = \ln n - (1 - \gamma) + o(1),$$

where H_n is the harmonic number $1 + 1/2 + \cdots + 1/n$.

In contrast, Lemma 8.13 shows that $\lfloor \log_4 n \rfloor \leq \mathtt{M}^\star(n) \leq \log_4 n$ (similar bounds appear in [CBFH$^+$97, BP19]).

We prove Theorem 8.15 by proving a lower bound and then a matching upper bound. We start with the lower bound.

Lemma 8.16. *Consider prediction using expert advice with n experts. For any proper learner Lrn_p there exists a strategy for the adversary under which the loss of the learner is at least $H_n - 1$. Consequently,*

$$\mathtt{M}^\star_p(n) \geq H_n - 1.$$

Proof. We will run the prediction game for $n-1$ rounds. At the i'th round, let G_i be the set of experts which are consistent with the examples seen so far, and let B_i be the remaining experts.

We set the true label to 0. All experts in B_i predict 1. An expert in G_i maximizing μ_i also predicts 1, and all other experts in G_i predict 0. Clearly $|G_{i+1}| = |G_i| - 1$, and the loss of the learner is

$$\mu_i(B_i) + \frac{\mu_i(G_i)}{|G_i|} = \frac{1}{|G_i|} + \frac{|G_i| - 1}{|G_i|}\mu_i(B_i) \geq \frac{1}{|G_i|}.$$

After $n-1$ rounds, there is precisely one expert left, and the loss of the learner is at least

$$\sum_{i=2}^{n} \frac{1}{i} = H_n - 1. \qquad \square$$

The matching upper bound is given by a natural "follow the leader" algorithm.

Lemma 8.17. *Consider prediction using expert advice with n experts. Let* FTL *be the algorithm which chooses a random expert among those who have not made any mistake so far. The loss of* FTL *on any realizable sequence is at most $H_n - 1$. Consequently,*

$$\mathtt{M}^\star_p(n) \leq H_n - 1.$$

Proof. As in the proof of Lemma 8.16, let G_i be the set of experts which have not made any mistake before round i. Thus $|G_1| = n$, and at all times, $|G_i| \geq 1$. The loss of FTL in the i'th round is precisely

$$\frac{|G_i| - |G_{i+1}|}{|G_i|} = \sum_{j=|G_{i+1}|+1}^{|G_i|} \frac{1}{|G_i|} \leq \sum_{j=|G_{i+1}|+1}^{|G_i|} \frac{1}{j}.$$

Therefore the total loss of the learner across all rounds is

$$\sum_{i=1}^{\infty} \frac{|G_i| - |G_{i+1}|}{|G_i|} \leq \sum_{i=1}^{\infty} \sum_{j=|G_{i+1}|+1}^{|G_i|} \frac{1}{j} \leq \sum_{j=2}^{n} \frac{1}{j}. \qquad \square$$

9 Open Questions

Our work naturally raises many directions for future work.

General Questions

Multiclass setting. Daniely et al. [DSBDSS15] extended the definition of Littlestone dimension to the multiclass setting, and showed that it gives the exact mistake bound for deterministic algorithm. Can we extend the definition of randomized Littlestone dimension to this setting?

A potential application is the problem of prediction using expert advice when the predictions are non-binary, a setting studied in [BP19].

For more recent work on multiclass classification which involves various combinatorial dimensions, see [BCD+22, KVK22].

Proper learning of arbitrary hypothesis classes. In Section 8.6 we show that improper learning algorithms outperform proper learning algorithm in online learning of the hypothesis class $\mathcal{U}_n$. What can we say about arbitrary hypothesis classes, and in particular, about the ratio $\mathtt{M}^\star_p(\mathcal{H})/\mathtt{M}^\star(\mathcal{H})$?

Mistake Bounds

Adaptive algorithms. Algorithm WeightedRandSOA gives the optimal mistake bound, but requires knowledge of k. Theorem 7.7 gives an algorithm which doesn't require knowledge of k, and has a regret bound of $\tilde{O}(\sqrt{\mathtt{M}^\star(\mathcal{H},k)\cdot\log k})$ (this is the loss beyond $\mathtt{M}^\star(\mathcal{H},k)$). What is the optimal regret bound?

Speed of convergence to the mistake bound. Given a hypothesis class $\mathcal{H}$, how many rounds are needed in order to guarantee a loss of $\mathtt{RL}(\mathcal{H})-\epsilon$? Proposition 6.3 shows (via Theorem 6.1) that the answer is at most $2\mathtt{RL}(\mathcal{H})+O(\log(1/\epsilon))$. Is this tight whenever $\mathcal{H}$ is infinite?

Proposition 6.9 shows that this bound is tight when $\mathcal{H}$ is "strongly infinite", and Proposition 6.7 gives a lower bound of $\log(1/\epsilon)$ for all infinite $\mathcal{H}$, leaving a gap of $2\mathtt{RL}(\mathcal{H})$.

A related questions concerns the regime in which the number of rounds is less than $2\mathtt{RL}(\mathcal{H})$. For every $\mathbf{T}$ it clearly holds that $\mathtt{RL}(\mathcal{H},\mathbf{T})\le\mathbf{T}/2$, and when $\mathbf{T}\le 2\mathtt{RL}(\mathcal{H})-\omega(\sqrt{\mathbf{T}\log\mathbf{T}})$, this is tight up to an $o(1)$ additive term, as the proof of Proposition 6.5 shows. For larger $\mathbf{T}$, the error term in the proposition gets larger, reaching $O(\sqrt{\mathbf{T}\log\mathbf{T}})$ for $\mathbf{T}$ close to $2\mathtt{RL}(\mathcal{H})$. What is the optimal bound on $\mathbf{T}/2-\mathtt{RL}(\mathcal{H},\mathbf{T})$ for the entire range $\mathbf{T}\le 2\mathtt{RL}(\mathcal{H})$?

Characterizing the equality cases of $\mathtt{M}^\star(\mathcal{H})\le\mathtt{M}^\star_D(\mathcal{H})\le 2\mathtt{M}^\star(\mathcal{H})$. In Section 5.3.2 we gave two examples showing that both inequalities can be tight. Can we characterize the two families of classes for which each inequality is tight? For example, it can be shown that every class $\mathcal{H}$ satisfying $\mathtt{M}^\star(\mathcal{H})=\mathtt{M}^\star_D(\mathcal{H})$ must be infinite, but not vice versa.

Prediction using Expert Advice

Proper predictions and repeated game playing. Consider the prediction using the expert advice problem, when the learner is restricted to predict with a convex combination of the experts. That is, at the beginning of each round (before seeing the advice of the n experts), the learner picks a convex combination of the experts and predicts accordingly. What is the optimal expected number of mistakes in this game?

We comment that this game can also be presented in the language of game theory: assume a repeated zero-sum game with 0/1 values, where each round is played as follows: player (i) chooses a (mixed) strategy and reveals it to player (ii), who then replies with a strategy of his own. What is the optimal accumulated payoff that player (i) can guarantee provided that she has n strategies and that the sequence of strategies chosen by player (ii) is such that player (i) has a pure strategy that loses to at most k of them? Proper predictions in the prediction with expert advice setting are equivalent to mixed strategies here.

Prediction using expert advice with different budgets. Section 8 considers prediction using expert advice in the k-realizable setting. The goal is to determine $\mathtt{L}_k(\mathcal{U}_n)$ and $\mathtt{RL}_k(\mathcal{U}_n)$. One can ask more generally for the deterministic and randomized Littlestone dimensions of the weighted hypothesis class $\mathcal{U}_{k_1,\dots,k_n}=\{(h_1,k_1),\dots,(h_n,k_n)\}$, where $h_1,\dots,h_n$ are the hypotheses in $\mathcal{U}_n$. In particular, which parameter determines the ratio $\mathtt{RL}(\mathcal{U}_{\vec{k}})/\mathtt{L}(\mathcal{U}_{\vec{k}})$?

In the case of two experts, the arguments in Theorem 8.11 can be extended to give an exact formula for both quantities:

$$\mathtt{RL}(\mathcal{U}_{k,\ell}) = \frac{k\binom{k+\ell+1}{\geq \ell+1} + \ell\binom{k+\ell+1}{\geq k+1} + (k+\ell+1)\binom{k+\ell}{k}}{2^{k+\ell+1}} \text{ and } \mathtt{L}(\mathcal{U}_{k,\ell}) = k+\ell+1.$$

Roughly speaking, $\mathtt{RL}(\mathcal{U}_{k,\ell}) \approx \max(k,\ell)$,[14] and so

$$\frac{\mathtt{RL}(\mathcal{U}_{k,\ell})}{\mathtt{L}(\mathcal{U}_{k,\ell})} \approx \frac{\max(k,\ell)}{k+\ell}.$$

Experiments suggest that more generally, if k_1, k_2 are the two largest elements in $\vec{k}$, then

$$\frac{\mathtt{RL}(\mathcal{U}_{\vec{k}})}{\mathtt{L}(\mathcal{U}_{\vec{k}})} \approx \frac{\max(k_1,k_2)}{k_1+k_2}.$$

Determining $\mathtt{M}^{\star}_{D}(n,k)$. Our work shows that $\mathtt{M}^{\star}_{D}(n,k)$ is the maximal depth m of a complete binary tree which is k-shattered by $\mathcal{U}_n$. If the tree is *uniform*, that is, all vertices at level ℓ are labelled by the same instance x_ℓ, then the tree is k-shattered by $\mathcal{U}_n$ if $\{(h(x_1),\ldots,h(x_m)) : h \in \mathcal{U}_n\}$ is a covering code for $\{0,1\}^m$ of radius k. Conversely, if such a covering code exists, then $\mathtt{M}^{\star}_{D}(n,k) \geq m$.

It is possible to obtain the lower bound $\mathtt{M}^{\star}_{D}(n,k) \geq D(n,k) - O(\log D(n,k))$ of [CBFHW96] using known constructions of covering codes [Str94, Lemma 4.1; CHLL97, Theorem 12.1.2]. The corresponding tree is uniform. Can we obtain better lower bounds on $\mathtt{M}^{\star}_{D}(n,k)$ using non-uniform trees, tightening the gap between $\mathtt{M}^{\star}_{D}(n,k)$ and $D(n,k)$?

Efficient implementation of WeightedRandSOA. In Section 8.5 we show how to implement RandSOA efficiently in the "perfect expert" setting. Can WeightedRandSOA be implemented efficiently on $\mathcal{U}_n$ for $k \geq 1$, say in time *poly*(n,k)?

[ALW06, BP19] observed that the only information relevant to the adversary's choice of expert predictions is the state of each round, which is indicated by a $(k+1)$-ary vector specifying how many experts have $i \in \{0,\ldots,k\}$ mistakes left. Using this observation, it is straightforward to derive an algorithm that calculates the randomized Littlestone dimension of every possible state in time complexity $O(n^{2k})$, and then uses these values to determine the optimal prediction in each round efficiently. Can we improve this?

References

[ABED+21] Noga Alon, Omri Ben-Eliezer, Yuval Dagan, Shay Moran, Moni Naor, and Eylon Yogev. Adversarial laws of large numbers and optimal regret in online classification. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 447–455, 2021.

[ABL+22] Noga Alon, Mark Bun, Roi Livni, Maryanthe Malliaris, and Shay Moran. Private and online learnability are equivalent. *J. ACM*, 69(4):28:1–28:34, 2022. `doi:10.1145/3526074`.

[ALW06] Jacob Abernethy, John Langford, and Manfred K Warmuth. Continuous experts and the binning algorithm. In *International Conference on Computational Learning Theory*, pages 544–558. Springer, 2006.

[14] This follows from the formula $\mathtt{RL}(\mathcal{U}_{k,l}) = 2\mathbb{E}[\max(\mathrm{Bin}(k+1,\frac{1}{2}), \mathrm{Bin}(\ell+1,\frac{1}{2}))] - 1$.

[BCD$^+$22] Nataly Brukhim, Daniel Carmon, Irit Dinur, Shay Moran, and Amir Yehudayoff. A characterization of multiclass learnability, 2022. URL: `https://arxiv.org/abs/2203.01550`.

[BDPSS09] Shai Ben-David, Dávid Pál, and Shai Shalev-Shwartz. Agnostic online learning. In *COLT*, 2009.

[BEHW89] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. Learnability and the Vapnik–Chervonenkis dimension. *Journal of the ACM (JACM)*, 36(4):929–965, 1989.

[BNS19] Amos Beimel, Kobbi Nissim, and Uri Stemmer. Characterizing the sample complexity of pure private learners. *Journal of Machine Learning Research*, 20(146):1–33, 2019. URL: `http://jmlr.org/papers/v20/18-269.html`.

[BP19] Simina Brânzei and Yuval Peres. Online learning with an almost perfect expert. *Proceedings of the National Academy of Sciences*, 116(13):5949–5954, 2019.

[CBFH$^+$97] Nicolo Cesa-Bianchi, Yoav Freund, David Haussler, David P Helmbold, Robert E Schapire, and Manfred K Warmuth. How to use expert advice. *Journal of the ACM (JACM)*, 44(3):427–485, 1997.

[CBFHW96] Nicolo Cesa-Bianchi, Yoav Freund, David P Helmbold, and Manfred K Warmuth. On-line prediction and conversion strategies. *Machine Learning*, 25(1):71–110, 1996.

[CBL06] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

[CHLL97] Gérard Cohen, Iiro Honkala, Simon Litsyn, and Antoine Lobstein. *Covering codes*. Elsevier, 1997.

[Cov65] T. Cover. Behavior of sequential predictors of binary sequences. In *Proc. of the 4th Prague Conference on Information Theory, Statistical Decision Functions and Random Processes*, pages 263–272. Publishing House of the Czechoslovak Academy of Sciences, 1965.

[DFGM17] Yuval Dagan, Yuval Filmus, Ariel Gabizon, and Shay Moran. Twenty (simple) questions. Online preprint, 2017. `https://yuvalfilmus.cs.technion.ac.il/Papers/frugal.pdf`.

[Doo53] J. L. Doob. *Stochastic processes*. John Wiley & Sons, Inc., New York; Chapman & Hall, Ltd., London, 1953.

[DSBDSS15] Amit Daniely, Sivan Sabato, Shai Ben-David, and Shai Shalev-Shwartz. Multiclass learnability and the ERM principle. *J. Mach. Learn. Res.*, 16:2377–2404, 2015.

[DSS14] Amit Daniely and Shai Shalev-Shwartz. Optimal learners for multiclass problems. In *COLT*, pages 287–316, 2014.

[Ele14] Neven Elezović. Asymptotic expansions of central binomial coefficients and Catalan numbers. *J. Integer Seq.*, 17(2):Art. 14.2.1, 14, 2014.

[Fel17] Vitaly Feldman. A general characterization of the statistical query complexity. In Satyen Kale and Ohad Shamir, editors, *Proceedings of the 30th Conference on Learning Theory, COLT 2017, Amsterdam, The Netherlands, 7-10 July 2017*, volume 65 of *Proceedings of Machine Learning Research*, pages 785–830. PMLR, 2017. URL: `http://proceedings.mlr.press/v65/feldman17c.html`.

[FX15] Vitaly Feldman and David Xiao. Sample complexity bounds on differentially private learning via communication complexity. *SIAM J. Comput.*, 44(6):1740–1764, 2015. `doi:10.1137/140991844`.

[Ham50] Richard W Hamming. Error detecting and error correcting codes. *The Bell system technical journal*, 29(2):147–160, 1950.

[Han14] Steve Hanneke. Theory of disagreement-based active learning. *Foundations and Trends® in Machine Learning*, 7(2-3):131–309, 2014. URL: `http://dx.doi.org/10.1561/2200000037`, `doi:10.1561/2200000037`.

[Haz19] Elad Hazan. Introduction to online convex optimization. *arXiv preprint arXiv:1909.05207*, 2019.

[HLM21] Steve Hanneke, Roi Livni, and Shay Moran. Online learning with simple predictors and a combinatorial characterization of minimax in 0/1 games. *Proceedings of Machine Learning Research*, 134:1–26, 2021.

[HY15] Steve Hanneke and Liu Yang. Minimax analysis of active learning. *J. Mach. Learn. Res.*, 16:3487–3602, 2015. URL: `https://dl.acm.org/doi/10.5555/2789272.2912111`, `doi:10.5555/2789272.2912111`.

[KLMZ17] Daniel M. Kane, Shachar Lovett, Shay Moran, and Jiapeng Zhang. Active classification with comparison queries. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 355–366. IEEE Computer Society, 2017. `doi:10.1109/FOCS.2017.40`.

[KvE15] Wouter M. Koolen and Tim van Erven. Second-order quantile methods for experts and combinatorial games. In *Proceedings of the 28th Conference on Learning Theory*, 2015.

[KVK22] Alkis Kalavasis, Grigoris Velegkas, and Amin Karbasi. Multiclass learnability beyond the pac framework: Universal rates and partial concept classes, 2022. URL: `https://arxiv.org/abs/2210.02297`.

[Lit88] Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine learning*, 2(4):285–318, 1988.

[LW94] Nick Littlestone and Manfred K Warmuth. The weighted majority algorithm. *Information and computation*, 108(2):212–261, 1994.

[MS10] Indraneel Mukherjee and Robert E Schapire. Learning with continuous experts using drifting games. *Theoretical Computer Science*, 411(29-30):2670–2683, 2010.

[Nat89] Balas K. Natarajan. On learning sets and functions. *Machine Learning*, 4(1):67–97, 1989.

[Ros58] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[Sha12] Shai Shalev-Shwartz. Online learning and online convex optimization. *Found. Trends Mach. Learn.*, 4(2):107–194, 2012. `doi:10.1561/2200000018`.

[SSBD14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[Str94] René Struik. *Covering Codes*. PhD thesis, Eindhoven University of Technology, 1994.

[VC74] Vladimir Vapnik and Alexey Chervonenkis. *Theory of Pattern Recognition*. Nauka, 1974.

[Vov90] Volodimir G Vovk. Aggregating strategies. *Proc. of Computational Learning Theory, 1990*, 1990.

[Wik22] Wikipedia contributors. Binomial coefficient — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Binomial_coefficient&oldid=1117456458`, 2022. [Online; accessed 22-October-2022].